NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DYNAMIC ASSEMBLY FOR SYSTEM ADAPTABILITY, DEPENDABILITY AND ASSURANCE (DASADA) PROJECT ANALYSIS
 	
by
 	
Wayne S. Mandak
and
Charles A. Stowell, II
 	
June 2001
 	
Thesis Advisor: Luqi
Co-Advisors: Man-Tak Shing
                               John Osmundson
                               Richard Riehle

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Dynamic Assembly for System Adaptability, Dependability and Assurance (DASADA) Project Analysis		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Wayne S. Mandak and Charles A. Stowell II			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT *(maximum 200 words)*

This thesis focuses on an analysis of the dynamic behavior of software designed for future Department of Defense systems. The DoD is aware that as software becomes more complex, it will become extremely critical to have the ability for components to change themselves by swapping or modifying components, changing interaction protocols, or changing its topology. The Defense Advanced Research Programs Agency formed the Dynamic Assembly for Systems Adaptability, Dependability, and Assurance (DASADA) program in order to task academia and industry to develop dynamic gauges that can determine run-time composition, allow for the continual monitoring of software for adaptation, and ensure that all user defined properties remain stable before and after composition and deployment. Through the study, a review of all the DASADA technologies were identified as well as a thorough analysis of all 19 project demonstrations.

This thesis includes a template built using the object-oriented methodologies of the Unified Modeling Language (UML) that will allow for functional and non-functional decomposition of any DASADA software technology project. In addition, this thesis includes insightful conclusions and recommendations on those DASADA projects that warrant further study and review.

14. SUBJECT TERMS Computing, Software, Software Engineering, Software Demonstration			**15. NUMBER OF PAGES** 90
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

DYNAMIC ASSEMBLY FOR SYSTEM ADAPTABILITY, DEPENDABILITY AND ASSURANCE (DASADA) PROJECT ANALYSIS

Wayne S. Mandak
Major, United States Marine Corps
B.S., Allegheny College, 1983

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

Charles A. Stowell II
Lieutenant Commander, United States Naval Reserve
B.S., The Citadel, 1985

Submitted in partial fulfillment of the requirements for the degree of

Master of Science in Information Technology Management
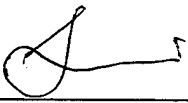
from the
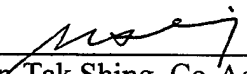
**NAVAL POSTGRADUATE SCHOOL
June 2001**

Authors: _____
Wayne S. Mandak
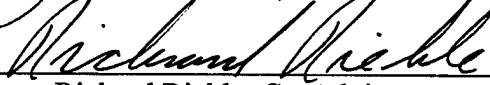
_____
Charles A. Stowell II

Approved by: _____
Luqi, Thesis Advisor

_____
Man-Tak Shing, Co-Advisor

_____
John S. Osmundson, Co-Advisor

_____
Richard Riehle, Co-Advisor

_____
Dan Boger, Chairman, Computer Science
And Information Systems Academic Group

iii

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis focuses on an analysis of the dynamic behavior of software designed for future Department of Defense systems. The DoD is aware that as software becomes more complex, it will become extremely critical to have the ability for components to change themselves by swapping or modifying components, changing interaction protocols, or changing its topology. The Defense Advanced Research Programs Agency formed the Dynamic Assembly for Systems Adaptability, Dependability, and Assurance (DASADA) program in order to task academia and industry to develop dynamic gauges that can determine run-time composition, allow for the continual monitoring of software for adaptation, and ensure that all user defined properties remain stable before and after composition and deployment. Through the study, a review of all the DASADA technologies were identified as well as a thorough analysis of all 19 project demonstrations.

This thesis includes a template built using the object-oriented methodologies of the Unified Modeling Language (UML) that will allow for functional and non-functional decomposition of any DASADA software technology project. In addition, this thesis includes insightful conclusions and recommendations on those DASADA projects that warrant further study and review.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENT

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PURPOSE

The primary focus of this thesis is to provide recommendations to the Program Manager of the Dynamic Assembly for Systems' Adaptability, Dependability, and Assurance (DASADA) program created by the Defense Advanced Research Projects Agency (DARPA) on the merits of new software engineering technologies and their possible integration with respect to future Department of Defense (DoD) systems. Recommendations will be based on an in-depth study of 19 separate technologies submitted to DARPA in response to a need for military software systems to be able to change themselves by swapping or modifying components and protocols dynamically while the system is operating. This thesis is intended to provide a thorough evaluation of all the technologies submitted by industry and research universities, standardizing the acceptance process, and submitting the results to DARPA.

This thesis will utilize the DASADA evaluation criteria developed by DARPA to ensure the technologies will perform the following criteria in order of importance: (1) Overall scientific and technical merit. Consideration was given to both the technology produced and the approaches used to ensure that the technology does (or can) produce the benefits claimed. (2) Understanding of problem and relevance of research effort to DASADA objectives. Evaluation of the projects was based on the extent to which they support dynamic assembly (or re-assembly) of components and on the specificity with which they defined "gauges" to assess properties of components and systems. The gauges needed to have the capability of assuring critical properties of "off-the-shelf" or

1

"open source" components with respect to the requirements of a given system. (3) Capabilities, related experience, and qualifications of proposed project personnel. Teaming was encouraged. (4) Stated contribution and relevancy to DoD application. Evaluation of the projects was based on the extent to which they showed a match between the technology/gauges they develop and DoD system requirements, where the requirements are relevant to a family of systems, as opposed to a narrow niche. (5) Cost realism/reasonableness or best value. The overall estimated cost to accomplish the effort needed to be clearly shown as well as the substantiation of the costs for the technical complexity described (DARPA, 1999). The recommendations and information presented will benefit future DoD software systems, as the need for adaptable software that can change themselves by modifying or swapping components, interaction protocols, or topology dynamically, while the system is in operation, will be the benchmark. The thesis includes an evaluation of the Managed Information and Network Exchange Router (MINER) program, a U.S. Space and Naval Warfare Systems Command (SPAWAR) Command, Control, Communication, and Intelligence (C^4I) application, which acts as a template for other DoD programs interested in including DASADA in their software systems.

B.   RESEARCH QUESTIONS

The main research question is to identify and recommend a standardized methodology for implementing DASADA technologies into DoD software systems. This design needs to consider the requirements of the software in terms of reliability, dependability, and adaptability without the degradation of operability and run-time performance. Promulgating a template will help to ensure standardization and serve as a metric for approval or disapproval of the implementation of the DASADA technology in

2

a specific software system. The template will diagram the software architecture, the system components, desired functionality, and logical relationship among components with respect to the DASADA technologies.

Additional questions addressed include:

- Are the sponsored projects meeting the DASADA Program objective, which is to develop dynamic gauges or measures of component composability or interoperability?

- Are the sponsored projects establishing a "reasonable" plan of product demonstration as well as product implementation on a limited fielding level?

- Does the DASADA program dictate the sponsored projects as mutually supporting?

- Are the sponsored projects actually going to develop a product/system, which will provide benefits to the DoD, or is it going to provide a "theoretical" solution to the stated program goal?

C.   OVERVIEW

The objective of the DASADA Program is to research, develop, and transition critical technology that will enable mission critical systems to meet high assurance, high dependability, and high adaptability DoD requirements. The vision is that there is (through design or recovery) a description of system architecture, a specification of critical properties, and requirements for change. DASADA technology will need to enable architecture refinement with guarantees that critical system properties will be assured through design rules that guide the selection, adaptation, and dynamic run-time assembly of appropriate system components. DASADA techniques will be required to enable the modification of distributed and heterogeneous systems, and needs to assure those properties of "off-the-shelf" or "open-source" components are adequate with respect to the requirements of a specified system.

DASADA adopts a three-faceted concurrent engineering paradigm for adaptable, dependable, and high assurance mission critical systems: Continual Design, Continual Coordination, and Continual Validation. In this model, components are selected or constructed, and customized and evaluated before (Continual Design), during (Continual Coordination), and after (Continual Validation) system assembly, and on-the-fly re-assembly, to ensure that they can and do operate together with the rest of the system, and its current context, within tolerated bounds. Continual Validation is particularly essential for assured applications because assurances that may have been met at initial system design time may not prove to be appropriate for field conditions, which are subject to rapid change while the system is running. Such applications typically cannot be "taken down" for long reengineering or enhancement cycles, but must be dynamically assembled in response to feedback from run-time gauges of functional and non-functional system properties.

Two necessary bases for all three facets of dynamic assembly are: (1) being able to precisely determine and usefully specify the room for variation in components and their composition, and (2) being able to measure that components fit, and continue to fit, together as system and context change, within functional and non-functional tolerances permitted by dynamically evolving system requirements. The measurement probes must be insertable into legacy as well as new components and compositions, and "displayable" to humans and automated agents as useful and quickly interpretable gauges to prevent inappropriate system assemblies and trigger re-assemblies promptly when needed (Milligan, 2001).

D.   SCOPE

The scope of this thesis includes: (1) an evaluation of the current industry proposals for DASADA; (2) recommendations on the feasibility of each proposal for future DoD system architecture development; (3) generating a template for future evaluations of the DASADA technologies; and (4) a recommendation and rationale for the acceptance or rejection of the DASADA technologies to the program manager.

E.   METHODOLOGY

A review of the DARPA functional requirements listed in the request for proposal as well as various DASADA briefs, white papers, periodicals, and other DoD on-line resources was conducted. In addition, analysis generated from the DASADA program conference held at the Naval Postgraduate School, Monterey, CA from January 31 - February 2, 2001 was completed. An in-depth analysis of the 19 DASADA technologies was conducted during the DARPA-sponsored demonstration held in Baltimore June 4-5, 2001. Those programs not ready for the next phase or deemed not relevant to future DoD software systems were eliminated from consideration.

F.   EXPECTED BENEFITS OF RESEARCH

Large, modern software applications, including DoD C4I systems, are constructed from custom and preexisting components from a variety of sources. Both the components and their organization with respect to each other must evolve over time as the result of new requirements, bug fixes, performance improvements, feature enhancements, and changes in their environments as the systems with which they interact change. An essential (but not sole) requirement for safely and predictably making these changes is knowing how the components use each other. This includes dynamic behavior that cannot

5

be captured in any way other than to observe the behavior of the system running in its normal operating environment (Milligan, 2001).

What DASADA technologies attempt to do are introduce gauges that collect, analyze, and present information about how deployed instances of distributed software actually interact, how this compares with the desired (specified) interaction patterns, how far the effects of changes can propagate and whether an anticipated action is likely to be safe, and to identify subtle differences between environments that might be the source of puzzling misbehavior. The results will be software gauges suitable for use in profiling applications constructed using a variety of important technologies such as Java, Dynamic Link Libraries (DLLs), Common Object Request Broker (CORBA), and Hypertext Transfer Protocol (HTTP). Tools will be developed to deploy these gauges to selectively collect information that is needed to diagnose particular problems, monitor the effect of recent reconfigurations, or to serve as inputs to other tools being used to plan or manage the evolution of a system. Such gauges are a necessary part of the feedback process needed for software evolution as envisioned by the DASADA program (Milligan, 2001).

Particularly critical to military systems is the need to make software changes predictably to ensure safety and reliability. DASADA technologies will build on previous technology efforts in the areas of: (1) Design, to assess the suitability of existing or new, off the shelf or automatically generated components for insertion in a system before assembly, allowing automated (controlled) assembly and on-the-fly transformations that produce predictable, safe systems; (2) Coordination, to assess the correctness of a composition operation during assembly, allowing reconfigurations to be

conducted safely across heterogeneous, distributed dynamic systems and; (3) Validation, to allow continual run-time validation of critical system properties.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND INFORMATION

A. DASADA PROGRAM GENERAL PROBLEM

Defense Advanced Research Project's Agency (DARPA) Dynamic Assembly for Systems' Adaptability, Dependability, and Assurance (DASADA) program's problem is that large systems, which are made up of numerous subsystems, are getting more complex. These systems have become more difficult to understand, build, operate, and evolve due to such causes as:

- Tighter integration, higher performance, interwoven concerns of system reliability, safety, and security
- Increased usage of COTS products which are "black box" components
- Economic necessity of using COTS backbones
- Ripple effects of changing single components or embedded systems

The solution is seen as on-the-fly system reconfiguration such as a system that it capable of:

- Gauging its own health in terms of performance and reliability
- Ability to perform rapid integration or reconfiguration while online
- Possess scalable mechanisms

DASADA Program's mission is to create a process and set of tools that assist in building and maintaining distributed systems. The requirement to possess the ability to assemble the components of a system, which will likely contain COTS products due to either economic constraints or technology leveraging, is currently being poorly addressed by the commercial world. But the increased level of complexity of the DOD supported systems critically requires this capability to assemble heterogeneous components or products in a "reasonably" predictable manner.

DASADA Program technology can be thought as providing new and enhanced Architectural Description Languages (ADLs) and tools as well as integrated design-time and run-time gauges for the purpose of modeling predicted as well as actual system behavior. The real promise of DASADA technology is a better understanding of the component level interactions that are rapidly becoming critical to the design, development, deployment, and lifecycle maintenance such as technology refresh of any large-scale distributed system. The ultimate goal of DASADA technology is to provide for the dynamic assembly of large-scale systems in a "reasonably" predictable manner. This goal of DASADA is in sharp contrast to the current state of affairs in the information technology development realm where the ad-hoc approach is the norm and thus very little assurance is provided that the modifications to any particular part of the system will not negatively impact the overall system performance or system reliability— this utter lack of predictability is unacceptable for modern military software applications (SchaferCorp, 2001).

B.   DASADA PROGRAM OBJECTIVE

In short, the DASADA Program objective is to develop dynamic gauges or measures of component composability or interoperability. How the DASADA Program will achieve this objective is by researching, developing, and transitioning critical technology that will enable mission critical systems to meet high assurance, high dependability, and high adaptability of DoD requirements. The vision is that there is through design or recovery, a description of system architecture, a specification of critical properties, and requirements for change. DASADA technology will need to enable architecture refinement with guarantees that critical system properties will be assured through design rules that guide the selection, adaptation, and dynamic run-time assembly

10

of appropriate system components. DASADA techniques will be required to enable the modification of distributed and heterogeneous systems, and needs to assure those properties of "off-the-shelf" or "open-source" components are adequate with respect to the requirements of a specified system.

DASADA adopts a three-faceted concurrent engineering paradigm for adaptable, dependable, and high assurance mission critical systems: Continual Design, Continual Coordination, and Continual Validation. In this model, components are selected or constructed, and customized and evaluated before (Continual Design), during (Continual Coordination), and after (Continual Validation) system assembly, and on-the-fly re-assembly, to ensure that they can and do operate together with the rest of the system, and its current context, within tolerated bounds. Continual Validation is particularly essential for assured applications because assurances that may have been met at initial system design time may not prove to be appropriate for field conditions, which are subject to rapid change while the system is running. Such applications typically cannot be "taken down" for long reengineering or enhancement cycles, but must be dynamically assembled in response to feedback from run-time gauges of functional and non-functional system properties.

Two necessary bases for all three facets of dynamic assembly are: (1) being able to precisely determine and usefully specify the room for variation in components and their composition, and (2) being able to measure that components fit, and continue to fit, together as system and context change, within functional and non-functional tolerances permitted by dynamically evolving system requirements. The measurement probes must be insertable into legacy as well as new components and compositions, and "displayable"

11

to humans and automated agents as useful and quickly interpretable gauges to prevent

inappropriate system assemblies and trigger re-assemblies promptly when needed.

User Space
* Develop probes (or probe specs)
* Modify arch specs (or rules)
* Develop/ specify gauges
* Deploy gauges / probes
* Specify adaptation processes /
rules

3) Scalable Event Infrastructure

Action

User

Reading

2) Monitoring and Analysis

Reading   Control

1) Measurement and Gauges

Sensor Data Control

4) Dynamic Adaptation

Architectural
Spec

Infrastructure
Adaptor

Target System

Activation
Filters
Patterns

Family of probes are
deployed into the system,
either passive or intrusive;
through wrappers or active
interfaces; in the
middleware or in the
components.

Figure 2.1.   DASADA System Architecture.

Figure 2.1 shows the proposed overall DASADA system architecture with the

four major system features:

- Measurement and Gauges

- Monitoring and Analysis

- Scalable Event Infrastructure

- Dynamic Adaptation

1.   Measurement and Gauges

The DASADA Measurement and Gauges objective is to provide a "gauge library"

to measure the approximate multi-dimensional fit of components with respect to

semantics and interaction behavior both at the design and run-time levels.

The DASADA Measurement and Gauges developmental approach has the

following features:

- Identify useful values for gauging system health and component fit
- Construct prototype gauges
- Integrate with standard communication infrastructures and evaluate the utility on real system problems

The DASADA Measurement and Gauges function is to demonstrate the ability to efficiently use these measurements and gauges in various system environments.

The DASADA Measurement and Gauges planned results are projects, which will demonstrate innovative indicators that provide:

- Structural and semantic measures to assess "approximate fit"
- Integration of semantics and dynamic architectural structure information
- Measures of time-varying configuration and usage in dynamic systems

An example of DASADA Measurement and Gauges is when design-time probes estimate the code and/or the time required for conversions and support selection of "best" route planner. As well as run-time probes which validate timing under different use conditions.

2. Monitoring and Analysis

The DASADA Monitoring and Analysis objective is to automate support to human as well as automated decisions about system restructuring or reconfiguring. This support will be non-intrusive, operate at multi-level granularity, and will assist in evolving the precision of the model.

The DASADA Monitoring and Analysis developmental approach is to demonstrate analyses based upon the comparison of gauge readings to structural, event, and ontological models as well as component "contracts". Additional comparative evaluations will be conducted with respect to utility and non-interference.

13

The DASADA Monitoring and Analysis planned results are projects, which will demonstrate the following capabilities:

- Architecture models created by ADLs which are linked to component "contracts"
- Pre-analyzed architecture parts used in predictive system models
- Use dynamically collected configuration and interaction information to determine the likelihood that a proposed software evolution is safe
- Incrementally refine models based upon its run-time monitoring

An example of DASADA Monitoring and Analysis is when monitoring and analysis routines use gauge measurements and design information, component contracts, as well as other forms of information to diagnose problems and plan repair strategies.

3. Infrastructure

The DASADA Infrastructure objective is to provide software generation capabilities to integrate gauges, analysis tools and adaptation mechanisms using underlying COTS (i.e. DLLs or XML) as well as standard representations (i.e. architecture or event sequences).

The DASADA Infrastructure developmental approach is to coordinate projects developing different integration frameworks such as HTTP or XML so that the components can interoperate. Additionally, all the technologies will demonstrate their interoperability in Technology Integration Experiments (TIE).

The DASADA Infrastructure function is to automate system, gauge, as well as analysis tools component composition.

The DASADA Infrastructure planned results are projects which will demonstrate the following capabilities:

- Deployment and configuration gauges

14

- Repair strategy specification language
- Composition risk assessment tool

An example of DASADA Infrastructure is Siena, which is a scalable internet-scale event notification service, which maintains effective mechanisms for selection and distribution of events based on interest such as "publish-subscribe". Clients can subscribe based upon any or all of the notification contents and/or patterns of events.

4. Dynamic Adaptation

The DASADA Dynamic Adaptation objective is to provide the ability to predictably and efficiently reconfigure systems on-the-fly based upon gauge readings and analyses as well as system models.

The DASADA Dynamic Adaptation development approach is to demonstrate dynamic system composition and gauge generation.

The DASADA Dynamic Adaptation function is to evaluate compatibility with COTS infrastructures.

The DASADA Dynamic Adaptation planned results are projects, which will demonstrate the following capabilities:

- Dynamically construct and reconfigure a concrete instantiation of a web-based architecture on-the-fly
- Reusable architectural transformations applied to evolving systems at run-time to increase system dependability

An example of DASADA Dynamic Adaptation would be the resultant reduction in effort to dynamically modify complex information management tasks with assured semantic and syntactic behavior (SchaferCorp, 2001).

C.   DARPA'S TECHNOLOGY DEVELOPMENT PROJECTS CRITERIA

Success in DARPA technology projects of which DASADA in one of many is dependent upon the following four primary criteria (SchaferCorp, 2001):

- Demonstration or proof that the new technique or technology works and is useful, where useful is defined as provides added value with respect to some defined capabilities

- Uses, as well as builds on, existing theory or technology—does not re-invent the wheel

- Provides evaluation results that are sufficient to convince someone to use it or develop it further

- Consistency with emerging standards such as component-based or uses commercial market standard communication infrastructures such as DCOM, CORBA, DCE, and XML

Additional DASADA evaluation criteria is required for all technologies to demonstrate:

- Predictable integration on new capabilities

- Reliable, automated adaptation of complex systems in the face of varying resources and user needs. Effective diagnosis and repair of real configuration and operational problems

- Guaranteed constraint satisfaction

- Improved throughput and response times for event driven and data driven applications

The following key technical issues must also be maintained:

- Heterogeneous computers and software infrastructures such as OS, languages, and resource allocation policies

- Rapid dynamic assembly of components

- Comprehensive system analysis such as timing, safety, and reliability

- Non-invasive instrumentation of a complex real-world software application such as SPAWAR/GDIS MINER Project

16

III. SPONSORED PROJECTS PRINCIPLES

DASADA Program is broken down into two phases. Phase one will focus on technology refinement and integration to provide on-the-fly system composition as well as recomposition that can adapt to new requirements while at the same time preserving the specified system critical properties. Phase two focuses on quantitative evaluations and further integration of DASADA technologies' ability to non-invasively instrument a variety of complex real-world software applications as well as to effectively diagnose and repair real configuration and operational problems in a number of systems.

The focus of this thesis is on phase one, so only the phase one performers are evaluated. The following information was obtained from the DASADA Program Project Information Sheets (Milligan, 2001). The breakdown of projects to area of technology is depicted in Table 3.1.

A. MESO-ADAPTATION OF SYSTEMS

Referring to Table 3.1 this project falls under the technology area of Measurement and Gauges. This project is a Georgia State University (GSU) effort with Melody Moore being the Project Lead Investigator. A major determinant of system reliability and safety is the degree to which the system's model of the social world (e.g. policies and doctrine governing decision-making autonomy and the ascription of significance to events) are compatible with the user's training and the "reality" of the organizational environment within which the system is embedded. Significant critical system failures have been traced to the occurrence of such ontological incompatibilities going unnoticed or unremarked, including the trivial incompatibility among units of measurement leading to

17

the loss of the Mars Climate Orbiter Probe in September 1999. Dynamic adaptation presents an even greater risk of these incompatibilities, as the opportunities for analysis and review are fewer. System components are compatible only to the extent that their ontology can be merged reliably.

Technology Area	Performer	Project
Measurement/Gauges	Georgia State University	Meso-Adaptation of Systems
	Kestrel	Specification-Carrying Software
	Object Services	Gauges to Dynamically Deduce Componentware Configurations
	SRI	Automated Dynamic Assembly of Dependable System Architectures
	Northrop Grumman	DACDLS
	University of Southern CA - ISI	TBASSCO
Monitoring/Analysis	BBN	Assured Assembly Infrastructure Toolkit
	Carnegie Mellon University	IMPACT
	Teknowledge	En-gauging Architectures
	University of Massachusetts	Process Guidance and Validation for Dependable On-the-Fly System Adaptation
	University of Oregon	Pacemaker
Infrastructure	Columbia/WPI	Coping With Complexity
	University of Colorado	Definition, Deployment, and Use of Gauges to Manage Reconfigurable Component-based Systems
	Carnegie Mellon University	Architecture-based Adaptation of Complex Systems
	University of Southern CA	Dynamic Assembly, Assessment, Assurance, and Adaptation via Heterogeneous Software Connectors
Dynamic Adaptation	Georgia Tech	DYNAMO
	Honeywell	Gauges for Reliable Adaptation
	University of CA - Irvine	Proteus
	Veridian - PSR	Innovative Gauges for Component-based System Assembly

Table 3.1.   DASADA Phase One Performers.

Meso-Adaptation is a form of software adaptation falling between the two extremes of macro-adaptation (major re-engineering) and micro-adaptation (run-time

18

tuning). In meso-adaptation, a change administrator makes changes to a system by configuring COTS/GOTS components that advertise their capabilities after subjecting them to computer-supported analyses of conceptual cohesion, compatibility and coverage (C4). These analyses yield quantitative estimates through MesoMorph's C4 gauges.

MesoMorph is a technology for evaluating the feasibility with which components can be integrated into existing systems. MesoMorph defines representations and adaptation gauges at these levels:

- Ontology: The system's implicit model of the world
- Context: Human capabilities, activity scenarios and situational factors in the assumed context of use
- Software architecture: Architectural adaptation wrappers for ontology and context. MesoMorph is a two-year effort involving a team of researchers at Georgia State University and Georgia Institute of Technology incorporating previous work by the investigators and other research in the DARPA community and beyond

For portability, all technology will be implemented with standard infrastructure for portability, including XML, UML, JLF/Swing, Java Beans and JINI. Two case studies will be performed. Examples of significant case studies are given in the body of the proposal and include environmental control and scenario-based planning of real-time battle simulations.

B.   SPECIFICATION-CARRYING SOFTWARE

Referring to Table 3.1 this project falls under the technology area of Measurement and Gauges. This project is a Kestrel Institute effort with the principal investigator being Dr Dusko Pavlovi. The focus of the project is on the composability of software systems, both at design-time and at run-time. The project is based upon the concept of specification-carrying software in which software artifacts carry with them all the information necessary to support composability and evolution. Kestrel will develop

19

techniques for measuring the compliance of a software artifact with its specification and provide measures at several levels of granularity, which will allow composability to be measured. The proposed finest-grain of measure is the specification of the glue-code necessary to fit the services of one component with the requirements of another.

The project concepts are embodied in the Evolutionary Programming Over Explicit Interfaces (EPOXI) system which builds on an advanced mathematical foundation to enable the design and evolution of large-scale, heterogeneous, distributed, time-critical systems. EPOXI along with the specification-carrying code will enable an innovative and powerful approach to gauges of composability. EPOXI will provide composability metrics that will determine:

- Exact Fit — components are immediately interface able with no undesirable consequences
- Tolerance Measure — no immediate exact fit, then EPOXI will measure precisely to what extent safety or other desired critical property margins will be affected
- Change Order — EPOXI can specify exactly what modifications are required to ensure that the selected set of critical properties are preserved
- Repair — EPOXI will be able to dynamically synthesize the necessary glue to assure fit to the desired tolerance

The guiding philosophy of EPOXI is refinement of requirement specifications into code that is correct by initial construction. It is the intent of the project to establish and preserve all required properties during the system refinement process. Additionally, those measured residual properties that cannot be established during design or assured during evolution will be translated into run-time monitors and related code to increase assurance.

C. GAUGES TO DYNAMICALLY DEDUCE COMPONENTWARE CONFIGURATIONS

Referring to Table 3.1 this project falls under the technology area of Measurement and Gauges. This project is an Object Services and Consulting, Inc. (OBJS) effort with the Program Lead Investigator being Dr. David L. Wells.

The proposed objective of this technology effort builds gauges to collect, analyze, and present information about how deployed instances of distributed software actually interact, how this compares with the desired interaction patterns, how far the effects of changes can propagate and whether an anticipated action is likely to be safe, and to identify subtle differences between environments that might be the source of puzzling misbehavior. The results will be software gauges suitable for use in profiling applications constructed using a variety of important technologies (Dynamic Link Libraries (DLLs), Common Object Request Broker (CORBA), Hypertext Transfer Protocol (HTTP)). Also anticipated are tools developed to deploy gauges to selectively collect information that is needed to diagnose particular problems, monitor the effect of recent reconfigurations, or to serve as inputs to other tolls being used to plan or manage the evolution of a system.

It is intended that gauges will be transparently attached to existing components or the pathways between them using existing interceptor technology. OBJS will identify the kinds of information that can be collected at the interception points and collect, manage, aggregate, and visualize the collected information. At a minimum, they will attempt to determine dynamic component connectivity, function/method invocation, timing, and exceptions. They will use this information to reconstruct both the instantaneous and

long-term behavior of monitored applications. They will then demonstrate using a test application that exercises the three interconnection activities (DLLs, CORBA, HTTP).

D. AUTOMATED DYNAMIC ASSEMBLY OF DEPENDABLE SYSTEM ARCHITECTURES

Referring to Table 3.1 this project falls under the technology area of Measurement and Gauges. This project is a SRI International effort with the Program Lead Investigator being Robert Riemenschneider. Anticipating that in the near future most systems will be constructed from pre-existing components, an infrastructure is needed to support a component-based lifecycle. Intercomponent communication mechanisms (CORBA, DCOM) and data interchange formats (XML, DOM), service discovery mechanisms (JINI, e-Speak), and even higher-level collaboration and delegation mechanisms (SRI's Open Agent Architecture) are several of the current emerging technologies.

A component-based lifecycle also poses new software engineering challenges. Most components developed for the commercial market will not be developed with the high dependability requirements of DoD mission-critical applications in mind. Therefore, the question that needs to be answered is: How can a dependable system be built from components that may not be dependable?

Basing systems on components will also increase the pace of system evolution. Components will quickly be declared to be obsolete and replaced by new versions. As new versions of components offering new capabilities become available, users will naturally want to exploit those capabilities. Another question to be answered is: How can dependability be maintained when a system is constantly evolving?

SRI, building on previous research, will attempt to answer these questions. Their research on the design and construction of architectures for secure distributed transaction

processing has shown how it is possible to build a secure system from not necessarily secure components. The primary innovation to their approach is to link an abstract architectural model that is proven secure to the implemented system architecture by a series of transformations that demonstrably preserve security. This link allows SRI to conclude that results obtained from their security analysis of the abstract model are applicable to the implementation as well. The same technique can be used to establish other system dependability properties.

SRI proposes to build upon their earlier research by:

- Making it easier to construct transformation chains that link abstract, analyzable system models to complex component-based system implementations by adding information to transformations about when they should be applied

- Using the links to dynamically update the abstract models as the running system evolves, making it possible to build system dependability gauges

- Introducing a capability to evolve the system architecture at runtime to improve dependability gauge readings, ensuring that functionality, performance, and dependability requirements will continue to be met as system components are added and replaced

E. DYNAMICALLY ADAPTABLE COMPONENT-BASED DATA LINK SYSTEMS (DACDLS)

Referring to Table 3.1 this project falls under the technology area of Measurement and Gauges. This project is a Northrop Grumman Corporation effort with the principal investigator being Dwight Cass. Project proposes an innovative gauge technology approach to develop component-based real-time avionics system capable of safe, accurate, and predictable in-flight dynamic reconfiguration. Project leverages a flight-tested B-2 data link demonstration platform and Northrop Grumman Corporation's extensive domain expertise to provide technology that will dynamically reconfigure on-board avionics while assuring compatibility of new software insertions and mission

23

viability. Northrop Grumman Corporation will identify, develop, and validate gauges along two major axes: composition and operation, both in terms of functional correctness and resource utilization. There will be four major classes of gauges developed:

- Assembly -- quantify the functional correctness of components

- Consumption -- predict the ability of a component collection to safely and accurately function

- Diagnostic -- monitor the extent to which each component operationally meets its functional specification

- Performance -- monitor the extent to which each component operates within its resource budgets

The gauges will guide the operation of a configuration strategy engine to control the major phases of system configuration. The configuration strategy engine work will focus on:

Development of techniques to discover or create wrappers that resolve behavioral differences between components

Development of architecture model driven recovery scenarios that allow the system to consider various repair and restart strategies rather than wholesale system reconfiguration

F. TEMPLATE-BASED ASSURANCE OF SEMANTIC INTEROPERABILITY IN SOFTWARE COMPOSITION (TBASSCO)

Referring to Table 3.1 this project falls under the technology area of Measurement and Gauges. This project is a University of Southern California Information Sciences Institute effort with the principal investigator being Robert Neches. The project proposes a set of mechanics that directly address the issues of adaptive composition sensitive to quality concerns. USC-ISI's approach helps software developers engage in guided, efficient searches and gauge-based evaluations of the set of alternative system

24

implementations that can be built with the components currently available to them. TBASSCO's tools will support intertwining composition and manual programming to iteratively build adapters for fitting components into a system when they are functionally satisfactory but suffer interface mismatches. TBASSCO helps the developer understand the tradeoffs of alternative implementations, and use records of decisions to generate run-time monitors that warn when the resulting system is being pushed outside its design envelope. USC-ISI's approach centers on tools for developing abstract system templates, which define a framework for exploring alternative system implementations by drawing from among candidate component sets for each function delineated in the abstract system. TBASSCO's use of semantic component descriptions such as functional compatibility and data equivalence go beyond component interfaces such as data types to provide better assurance of compatibility. Additionally, a language of qualifiers on the component software descriptions that supports qualitative evaluation during the composition process enhances the semantic component descriptions. Formal architectural level views of system execution are provided which in turn provides an easier way to calculate resource usage and analyze performance.

TBASSCO enable developers to evaluate components' functional and data equivalence compatibility, find pertinent data conversion mappings, and predict performance of a component architecture under specific usage situations and hardware/networking environments. Once a system is composed users will be able to deploy run-time monitors to watch for constraint violations, detect bottlenecks, and gather data to improve performance estimations.

G. ASSURED ASSEMBLY INFRASTRUCTURE (AAI) TOOLKIT

Referring to Table 3.1 this project falls under the technology area of Monitoring and Analysis. This project is a BBN Technologies effort with the primary investigator being Nathan Combs. The intent of this project is to develop an Assured Assembly Infrastructure (AAI) Toolkit, which will realize dynamically composable systems based on specified performance objectives. The AAI Toolkit will provide uniform assembly model for heterogeneous system components, including gauges that measure and drive the dynamic assembly and reconfiguration of the software architecture. AAI Toolkit will be able to dynamically adapt system architectures to optimize system performance with respect to multi-dimension objective functions such as speed, accuracy, and efficiency.

AAI Toolkit uses a dynamic assembly mechanism for constructing software architectures of components and gauges. XML is used to bridge multiple-levels of description such as metadata and architecture and provide a fast and flexible XML data binding implementation, which will allow a designer to efficiently recompose architectures dynamically while preserving a scalable model description. To achieve these results the BBN Technologies will develop the following AAI Plugins as part of the AAI Toolkit:

- Advocates -- domain specific adapters that understand system requirements and components dependencies
- Assured Assembly Machine (AAM) -- interact with Advocates to perform requirements tradeoffs and produce assembles of Components and Gauges
- Executors -- implement the Architecture Model to realize the specified software system in the appropriate implementation model
- Gauges -- provide constant feedback to the AAM that composes/reconfigures the system to better match the Architecture Model requirements
- Software Components -- software or devices/processes that provide services for use by software

26

BBN Technologies will leverage existing technologies such as XML, Quick, JINI/Java, QuO, and ALP. The AAI Toolkit will emphasize a number of capabilities such as the use of distributed and varied components, the assembly of software architectures from components, and the dynamic modification of the architecture from gauge feedback.

H. IMPACT: INTEGRATED METHODS FOR PREDICTIVE ANALYTIC COMPOSITION AND TRADEOFF

Referring to Table 3.1 this project falls under the technology area of Monitoring and Analysis. The project is a team effort, including members from Carnegie Mellon University (CMU), the CMU Software Engineering Institute (SEI), and Lockheed Martin Tactical Aircraft Systems. The Project Lead Investigator is John Lehoczky from CMU.

The project objective is to demonstrate that predictable dynamic assembly of software systems from "software parts" is achievable. It will do so in the following manner:

- Develop a framework and methodology, analytical composability (AC), to compose analyzable models from sub-models using formal rules

- Predict multiple dimensions of system performance (e.g. real-time predictability and reliability) from "gauge values" of software parts

- Formalize design tradeoffs of system-wide properties using the design space of software parts

- Validate AC predictions in an instrumented runtime environment

The IMPACT project ideally would develop four integrated thrusts to build the AC framework:

- Develop, collect and catalog a broad set of pre-analyzed architectural patterns with associated gauges

- Construct an analytical framework within which one can combine the analytical models thereby composing software parts into assembled systems and use the identified gauges to predict system level attributes

27

- Develop a set of design decision aids, based on the Q-RAM modeling framework, but will implement ideas from multi-attribute utility theory to allow a tradeoff analysis to be conducted

- Create instrumented run-time support, which will offer fault- tolerance protection against erroneous behaviors from part composition and measure the empirical performance of the system to validate the results from the analytic framework

I. EN-GAUGING ARCHITECTURES

Referring to Table 3.1 this project falls under the technology area of Monitoring and Analysis. This project is a Teknowledge Corporation effort with Robert Balzer and David Wile as the Project Lead Investigators.

Teknowledge will create the infrastructure to design and deploy gauges on real distributed systems running on commercial platforms to monitor their architecture and measure their performance. This dynamic system information will be collected in a repository, made available to a wide variety of subscribers both automated and human, and used to validate performance, resource requirements, and other selected service qualities and to augment the system's robustness and responsiveness.

Early computing applications were so starved for memory and precious processor time that every detail used in their construction was "compiled away" if it did not directly affect functionality; in fact, such systems performed well in only very tightly-constrained contexts. Modern systems, lacking the extreme resource constraints of old, need not be as highly tuned to the precise usage context, thereby retaining the potential for robustness and adaptability. Modern systems benefit from two adaptive technologies:

- The ability to compose systems from reusable modules developed and compiled separately

- The ability to distribute computing processes onto autonomous computing nodes

28

Although these technologies enable the potential to adapt performance to widely varying contexts, much of the information important for such performance adaptation is still "compiled out" of modern systems.

Fortunately, determining when and how to adapt a running system to varying configurations and performance demands -- the "Quality of Service (QoS) demands" -- can be separated from system functionality. To obtain such information it is necessary to model a system's nominal behavior and compare it to its actual behavior for the system's current configuration. While these models are by nature incomplete, they are adequate for validating and tuning performance. Whenever the system deviates from the model, either the system must be reconfigured to achieve its QoS demands or the resources reapportioned to balance those demands. Modeling the system's nominal behavior enables these validations and adaptations to be separated from the system's functionality and to be supported by an external infrastructure.

Teknowledge's will attempt to build that validation and adaptation infrastructure by developing and deploying the gauges that track the system's dynamic architecture and measure its performance. They will also attempt to build on their experience with the Acme architecture description language and its Instrumented Connector technology (both developed under DARPA's EDCS Program) to monitor the actual run-time architecture of a system, to reify it into an architecture model repository, and to publish event notifications to "subscribers" interested in such changes to the architecture. Such subscribers comprise analyzers to determine whether dynamic system constraints are satisfied, simulators to establish the system's nominal behavior benchmark, trackers to

respond to differences between nominal and actual, and even GUI animators, potentially evoking a human response to redirect system resources.

Teknowledge will also build on their expertise in integrating DARPA's Quorem QoS Condition Service (QCS), and their Instrumented Connector technology, to provide the infrastructure that enables application designers to design and deploy the gauges needed to measure and validate the running system's performance. Using their Composability Framework Services technology, application engineers will then be able to decide how and when to use this performance and configuration information for adaptation to affect the QoS demands.

J.   PROCESS GUIDANCE AND VALIDATION FOR DEPENDABLE ON-THE-FLY SYSTEM ADAPTATION

Referring to Table 3.1 this project falls under the technology areas of Monitoring and Analysis. This project is a University of Massachusetts (UMASS) effort with the Project Lead Investigators being Leon Osterweil and Lori Clarke.

UMASS proposes to develop, demonstrate, and evaluate key technologies that support a revolutionary approach to nimbly adapting software systems on-the-fly yet also provide unprecedented dependability assurances. Such adaptation support will enable the coming generation of DOD embedded software systems to respond to new requirements or unforeseen circumstances in seconds or minutes, rather than months or years, as is currently the case with more traditional development methods.

On-the-fly adaptation carries the risk that incorrect adaptation may cause the system to become inoperable. UMASS therefore proposes a disciplined adaptation approach, centered on a description of the system's architecture, a repository of candidate components for substitution into instantiated configurations, and quantitative measures

30

(gauges) of the degree to which running systems and proposed enhancements conform to critical properties. They propose that responsibility for adaptation reside in an adapter component logically separate from the application system being adapted. The adapter consists of a precisely defined, and demonstrably effective executable process that directs the on-the-fly adaptation according to the architectural description, the available components in the repository, and a comprehensive suite of analyzers able to quickly and accurately compute the readings of the gauges that guide this adaptation.

The proposed project centers on the development, demonstration, and evaluation of two technologies central to this disciplined adaptation approach, a process definition and execution language, called Little-JIL, and a static data flow analysis system, called FLAVERS. Prof. Osterweil and his team under the DARPA EDCS project have developed little-JIL, where it has been used successfully to define processes in domains such as software development, electronic commerce, and robot coordination. These experiences suggest that Little-JIL, with modest modifications, can be used to precisely define the on-the-fly adaptation processes required here. They will attempt to evaluate this hypothesis by using Little-JIL to implement example adaptation processes. They will assess the effectiveness of both the language and the processes by measuring such properties as process size, speed, clarity and complexity, enhancing both language and process as experience dictates. In addition, they will attempt to explore the feasibility of using this technology to implement self-adaptation of the adaptation process itself.

Professor Clarke and her team under the DARPA Arcadia and EDCS projects, where it has been used to verify diverse properties of concurrent software systems written in Ada and a subset of Java, as well as architecture descriptions and Little-JIL process

31

programs, have developed FLAVERS. Experience with FLAVERS suggests that it can be use to analyze the components needed for on-the-fly composition, the architecture descriptions used to guide the selection of candidate configurations, and the candidate configurations themselves to assure that each conforms to specified critical properties. They propose to evaluate this hypothesis by analyzing example components and architecture specifications written in example architecture description languages (ADLs). UMASS will assess FLAVERS effectiveness by measuring the time and space required to perform its analyses, the number and types of properties that it conclusively evaluates, and the number of constraints that it generates for dynamic monitoring.

UMASS proposes to make their technologies and research results widely available, especially within the DASADA research community, through papers, presentations, demonstrations, and evaluation copies of software prototypes. Their technology nicely complements the architecture composition and real time analysis capabilities being developed by the Honeywell Technology Center and they have proposed an option to explore this integration further. UMASS will also continue to pursue transition opportunities with such organizations as US Army TACOM, Boeing, Mitre, Motorola, and General Dynamics.

K.   PACEMAKER: CONTINUOUS VALIDATION OF COMPLEX SYSTEMS

Referring to Table 3.1 this project falls under the technology area of Monitoring and Analysis. This project is a University of Oregon effort with Michal Young as its Project Lead Investigator.

Lacking comprehensive, precise models of complex dynamic systems, one must treat models based on available information as hypotheses about actual system structure

and behavior. When properties of a change have been verified using a model, the model becomes a set of assumptions whose violation invalidates the verification argument. This implies that analysis and verification of models must be integrated with continual system monitoring, both to evolve models along with systems and to detect unanticipated effects of changes. Addressing this challenge, the primary objective of the proposed Pacemaker project builds upon results of earlier DARPA-sponsored research to support continuous validation. The key technologies to be developed and evaluated are:

- Requirements monitoring gauges that continuously evaluate required properties, including quality-of-service. This is an extension of technology developed in the DARPA Quorum program

- Flexible synthesis of models from multiple sources of information. This is an extension of technology developed in the DARPA EDCS program, drawing also from the Assert project of the DARPA Quorum program

- Repurposing standard model-checking technology to treat (partial) architectural models as hypotheses that can be validated against observations, in addition to their more conventional use to verify that proposed compositions preserve critical system properties. This will be based partly on related technology from the Assert project, drawing also from the Perpetual Testing project of the DARPA EDCS program

- Dynamic checking of user-specified object protocols. These protocols subsume the connector protocols that can be specified and statically checked in architectural description languages, and can therefore be used to enforce architectural constraints or to check assumptions expressed in a model (which, in a dynamic system, may come to the same thing).

The Pacemaker project will contribute to a radical acceleration of the cycle by which a developer can pose and answer specific questions about a potential integration of components in a complex system, and provide a "backstop" of continued monitoring after deployment to compensate for the incompleteness and imprecision of knowledge about complex and dynamic software systems.

L. COPING WITH COMPLEXITY: A STANDARDS-BASED KINESTHETIC APPROACH TO MONITORING NON-STANDARD COMPONENT-BASED SYSTEMS

Referring to Table 3.1 this project falls under the technology area of Infrastructure. This project, also referred to as Kinesthetics eXtreme (KX), is a Joint Columbia University and Worcester Polytechnic Institute effort with the primary investigator being Gail Kaiser. The project objective is to provide an architecture-based approach to run-time monitoring (i.e. continual validation) of the dynamic functional and extra-functional properties of component-based systems. The technical basis on KX is that architectural models show how to develop testing regiments for verifying that components behave as expected during dynamic system evolution, integration, and re-configuration.

How the system works is that the target system has its architecture defined using an ADL and then this architecture is inspected by KX by semi-automatically inserting software probes into component ports and actualized connector middleware or wrappers. The inserted probes detect and report system events that cross component boundaries. Required and prohibited properties are defined as complex patterns over partially ordered sequences (POSETs) of system events. These complex patterns will be recognized as they either occur or by the their omission as the target system executes. This system behavior can either be represented as binary gauges or as sophisticated gauges that provide contextual information about anomalous conditions. These gauges can be integrated with either automated decision facilities or directly displayed in a human-oriented GUI.

The probes provide entry points to an orthogonal monitoring meta-architecture, which is superimposed upon the target system's architecture, while the connectors operate as active connectors. The events generated by the probes are converted to smart events, which are represented in XML. The meta-architecture is extensible and supports sophisticated gauges that may acquire at run-time XML processing modules for specific tag sets (markup tags used to indicate how to process specific POSETs to enable the manage and update gauges). This dynamic nature enables new gauges to be defined, represented, and acted upon while the system continues to run and thus no downtime or significant reconfiguration just for the purpose of retrofitting the monitoring infrastructure.

M. DEFINITION, DEPLOYMENT, AND USE OF GAUGES TO MANAGE RECONFIGURABLE COMPONENT-BASED SYSTEMS

Referring to Table 3.1 this project falls under the technology area of Infrastructure. This project is a University of Colorado effort with the principal investigator being Alexander Wolf. The project proposes to design, develop, and prototype a framework for managing the reconfiguration of distributed component-based systems. The framework is called FIRM, which stands for Framework for Interoperable Reconfiguration Measures. FIRM is founded on the definition of a set of novel gauges to assess a wide range of critical system properties, and a scalable infrastructure to mange both the deployment and use of gauges throughout an enterprise. FIRM addresses the DASADA objective of Continual Coordination by ensuring that reconfiguration-related interoperability problems are detected and mitigated at multiple points in the lifecycle of a system. University of Colorado's existing Software Dock, Menage, and Siena research projects provide the technical underpinnings of FIRM.

FIRM's set of novel gauges is capable of evaluating system configurations with respect to important interoperability properties. The set of gauges will measure:

- Consistency and inconsistency of configurations
- Actual configurations adopted by systems
- Properties across all possible configurations of a system
- Redundancy and reuse properties of systems
- Predict the costs of moving from one configuration to another

The gauge-based evaluations can be performed statically on the configuration specifications as well as on the deployed configurations and performed dynamically on executing systems. The project additionally provides the necessary infrastructure to effectively deploy and use gauges other than their own; as well as the means to deploy, activate, and replace components, to apply gauges for coordination, to insert gauges into activated systems, and to capture, fuse, and disseminate the outputs of gauges.

N.   ARCHITECTURE-BASED ADAPTATION OF COMPLEX SYSTEMS

Referring to Table 3.1 this project falls under the technology area of Infrastructure. This project is a Carnegie Mellon University effort with the primary investigator being David Garlan. The project objective is to reduce the cost and improve the reliability of making changes to complex systems by developing a new technology supporting automated, dynamic system adaptation via architectural models, explicit representation of user tasks, and performance-oriented run-time gauges. This technology will based upon three critical areas of innovation:

- Detection – the ability to determine dynamic (run-time) properties of complex, distributed systems through the use of probes that will collect status and performance information for networks and endpoints. In addition, it will also determine properties through mechanisms that will aggregate the results of multiple probes and combine them into values of performance-oriented gauges which will be rendered in application-architecture terms

36

- Resolution – the ability to determine when observed systems properties violate critical design assumptions. By maintaining an explicit run-time representation of a system's architectural design and its invariants. As well as by maintaining an explicit run-time representation of the users' task state which will capture the high level requirements which are imposed on the running system

- Adaptation – the ability to automate system adaptation in response to violations of design assumptions. By providing a rule-based mechanism that associates invariant violations with "repair strategies". Additionally, using style-based analysis techniques they will be able to verify that certain classes of rewrite strategies provably maintain or reestablish key architectural properties. As well as a new theory and set of tools, which support compositional creation of connectors, will enable a user to rapidly create new connectors with varying QoS properties. The tools will automatically generate "glue" code for component integration and interaction

The above stated capabilities will provide both (a) ability to handle system changes with respect to the specific performance-oriented gauges which is supported by their technology, and (b) extensible framework to handle additional gauges and system adaptation strategies produced by other DASADA projects. The vision is that these capabilities will dramatically reduce the need for user intervention in adapting systems to achieve quality goals, improve dependability of changes, and support a whole new breed of systems that can perform reliable self-modification in response to dynamic changes in the system environment.

O.   DYNAMIC ASSEMBLY, ASSESSMENT, ASSURANCE, AND ADAPTATION VIA HETEROGENEOUS SOFTWARE CONNECTORS

Referring to Table 3.1 this project falls under the technology area of Infrastructure. This project is a joint University of Southern California Center for Software Engineering and Lockheed-Martin Corporation effort with the principal investigator being Barry Boehm. The proposed dynamic assembly technology builds on USC's and other's architectural component mismatch capabilities to provide gauges indicating the particular type, dimension, subdimension, and value of the mismatch. This

then maps into USC's taxonomy of software architectural connectors for which there already exists partial mappings from the mismatches into the most effective classes of connectors, such as procedure call or event, which are likely to resolve the mismatch. The project will extend the current gauges, classes, and mappings based on a problem-driven set of priorities. Additionally, the project will extend the current SAAGE (integrated environment for transforming UCI's C2-style architectures--hierarchical network of concurrent components linked together by connectors--into UML) architecture framework for rapid dynamic composition and assessment as well as verification to ensure that the selected connectors are appropriately configured and dynamically integrated into the operational system. USC/Lockheed-Martin's approach identifies five types of gauges, which will be developed:

- Measure the functional suitability of a partially modeled component to an architecture:
  - Interface match
  - Behavior match
  - Interaction match
- Determining design-time and integration-time development risks based on the non-functional properties of interacting components
- Measure the C3 properties between and within heterogeneous semantic models:
  - Consistency between static (invariants and pre-/post-conditions) vs. dynamic (state charts)
  - Conformance of architecture to design to ensure valid refinement
  - Completeness of architecture
- Measure different aspects of new components versions:
  - Correctness of the new version wrt to the old version
  - Performance of the new version wrt to the old version
  - Robustness and reliability of a new component version

38

- Measure shared properties of heterogeneous connectors:
  - Throughput
  - Load
  - Security
  - Reliability

P.   DYNAMO: DYNAMIC ASSEMBLY FROM MODELS

Referring to Table 3.1 this project falls under the technology area of Dynamic Adaptation. This project technology is being developed by the Georgia Institute of Technology. The Project Lead Investigator is Dr. Spencer Rugaber.

The purpose of this technology is to develop automated composition of software systems in such a way as to guarantee various properties such as correctness, reliability, and resource utilization. Software system components may be Commercial Off-The-Shelf (COTS) or custom-built for military applications. The composition may take place statically, when the system is first configured, or dynamically, as new components are added or old components are replaced.

DYNAMO attempts to address this by making extensive use of declarative models of the software components. Models may be built from scratch for new components or derived by analyzing existing components. Models are abstract and therefore easier to maintain than software built by hand. Models enable automatic generation of software with guaranteed properties. Additionally, the same model may be used to construct the system and to gauge its performance when it runs. DYNAMO technology hopes to take advantage of these modeling properties.

The Project Lead Investigator plans on the following DYNAMO deliverables:

- Modeling notations for specifying system components
- Automatic code generators for building components from models

- A system composer supporting the static and dynamic composition of components
- An evaluation framework in the form of gauges that measure system properties

The intent is to use DYNAMO to build an operations planning environment that includes heterogeneous information sources, multiscale visualizations, and severe robustness requirements.

Q.   GAUGES FOR RELIABLE ADAPTATION

Referring to Table 3.1 this project falls under the technology areas of Measurement and Gauges as well as Dynamic Adaptation. This project is a Honeywell Technology effort with the Project Lead Investigator being Dr. Christopher W. Geib.

This project objective builds on existing work in Architecture Description Languages (ADLs), developing four new technologies to capture and reason the ways in which system components can be combined and adapted. It is envisioned that an integrated design and on-line adaptation process will be developed in which:

- Constraint-Based Gauges will capture critical constraints on component behaviors, I/O, and other compatibility restrictions (e.g., I/O attribute type constraints, attribute bounds, platform operating system, processor, memory or peripheral requirements)
- The UNiversal Constraint Language and Engine (UNCLE) will reason about constraint-based gauges to detect compatibility violations
- Real-Time Performance Gauges that will capture measures of component performance that affect composability (e.g., queue lengths, runtimes, latencies)
- Run-time Configuration Triggers that will respond to gauge readings by triggering tailored runtime reconfigurations or design revisions to correct problems and continuously improve system performance

It is expected that these new technologies will allow a system designer to rapidly and efficiently combine system components that have been annotated with gauges. The gauges will measure a broad variety of component aspects and performance features to

ensure compatibility and compliance with overall system requirements. Also, gauge-enhanced system components will be executable in an adaptive software environment where runtime gauge feedback is used to evaluate system performance and trigger dynamic adaptation of the system via component reconfiguration.

R. PROTEUS: ASSESSMENT AND ADAPTATION THROUGH DYNAMIC ARCHITECTURE TECHNOLOGY

Referring to Table 3.1 this project falls under the technology area of Dynamic Adaptation. This project is a University of California, Irvine (UCI) effort with Richard Taylor as the Project Lead Investigator.

Previous DARPA investment in software architecture research at UCI yielded key technical foundations for effective software reuse and dynamic application adaptation. UCI's primary objective is to leverage this investment and advance the technology, providing comprehensive support for application/component assessment, adaptation, and run-time change. They also plan to carry this work into the domain of real-time and fault-tolerant systems.

Architecture-based system development is central to their approach. Strict separation of an application into components (loci of computation) and event-based connectors (loci of communication):

- Provides a demonstrated, effective basis for run-time dynamism: their architectural models reside with the implementation, providing the key resource for assessing, planning, and effecting change

- Enables a variety of "wrapping" technologies to be used to adapt components to unanticipated uses

- Fosters the use of run-time monitors, within connectors, to dynamically assess system functioning

UCI will produce gauges, prototype tools, and an open, standards-based environment for supporting DASADA. They plan to develop gauges for assessing

41

component/application adaptability based upon a concept of "open points", monitoring real-time events, checking architectural constraints, assessing conformance of code to architecture, and others. Practical application of the gauges, support for creating adaptive applications, and mechanisms for effecting run-time change will be provided through a comprehensive, open, architecture-based application engineering environment and implementation frameworks. Openness will come from its architecture and its use of an XML-based standard for architecture information exchange (xADL). COTS development tools will be integrated within the environment. It will be used reflexively to support its own evolution, ensuring that UCI will provide a comprehensive set of usable functionality. Provision of application development frameworks, which include COTS technologies, will facilitate rapid development and support dynamism. Their scope also includes the application of configuration management techniques to the problem of run-time change, organizing and streamlining run-time changes into a traceable and accountable process that adheres to adaptive constraints specified at design-time.

UCI's work will also address real-time and fault-tolerant systems. Lockheed Martin (Owego, NY) intends to supply them with HARDPack, a commercial, real-time, fault-tolerant ORB and platform. UCI will use HARDPack to create an application development framework supporting dynamic, real-time, fault-tolerant applications. HARDPack will be utilized as a connector technology, and will also enable them to monitor events in real-time, supporting assessment.

Evaluation of the work will be supported in part by use of a realistic test bed. Through their partnership with Lockheed Martin they will experiment with their technologies either using software from an AWACS Advanced Technology

Demonstration project in which Lockheed Martin participated, or a Lockheed Martin flight control system from the DARPA DSSA/ADAGE program.

S. INNOVATIVE GAUGES FOR COMPONENT-BASED SYSTEMS ASSEMBLY

Referring to Table 3.1 this project falls under the technology area of Dynamic Adaptation. This project is a joint Veridian Pacific-Sierra Research (PSR) and Carnegie Mellon University effort with the principal investigator being John Paul Parker. The project proposes to develop a fully demonstrable, web-based composable systems environment and gauge test bed. The project objective is to design and develop the gauges required to assure system flexibility, robustness, and functionality as well as demonstrate them in a real systems context. The project proposes to research and develop a "Gauge Box" which will provide:

- Design gauges – syntax checker, syntax mismatch, and infrastructure compatibility

- Coordination gauges – semantic fit measurer, protocol analyzer, system suitability, data compatibility, and performance analyzer

- Validation gauges -- performance analyzer, model analyzer, system suitability, user constraint measurer, and data compatibility

Veridian-PSR and CMU propose to develop this "Gauge Box" which will deliver the following capabilities:

- Measure syntactic and semantic suitability of components in an architectural instance

- Measure goodness of fit to allow the insertion of more types of components which leads to greater system flexibility

- Measure aggregate fit of a collection of components working together in a system context

- Enable the user to override a failed match and use a component that would otherwise not be available during the coordination and assembly phase

- Permit the continual validation of a run-time system

The Veridian-PSR led effort will build upon the synergy of several key leading edge composable systems efforts, such as the marriage of Veridian-PSR's Venice, CMU's Acme, and Sun's Java technologies. Veridian-PSR will adapt its Venice web-based component assembly and experiment framework to utilize CMU's Acme architecture interchange language and tools.

Veridian-PSR and CMU's approach will allow for the research, development, testing, and demonstration of a series of component gauges over a network using a standard web browser. These gauges will be used to help dynamically reconfigure a distributed system using real C4ISR software components. The proposed demonstration builds upon the previous Veridian-PSR research that demonstrated a "warm swap" capability which is the ability to assemble C4ISR software components into a fusion application and dynamically swap components in order to reconfigure the application at run-time without system rebuild. The resultant demonstration will demonstrate the power of dynamic, composable systems, and gauges by showing measurable order of magnitude improvements over the current design and integration paradigms.

IV. CASE STUDY AND TEMPLATE CONSTRUCTION

A. BACKGROUND

The best method of analyzing how DASADA technologies would be applied to the realm of military software would be to conduct a case study of one of the several EDP programs. These programs offered the DASADA technologies to apply their wares to functionally enhance their program capabilities as well as to provide a test-bed for the DASADA technologies. The Managed Information and Network Exchange Router (MINER) program, which is a C4I system jointly developed by SPAWAR Systems Center and General Dynamics Information Systems (GDIS) was chosen as the test bed.

MINER is a policy-based information management tool set that provides for awareness, access, and delivery of near real-time information to tactical applications and end-users based upon evolving needs. MINER's goal is to help upper level decision makers gain better understanding faster through the development and usage of software components that collectively form a highly reusable framework for producing information analysis, organization, and representation applications. MINER provides an integrated user interface to assist in its managing an ad-hoc access to information either locally, over SIPRNET, or over GBS/Split-IP services.

The desired end state of the DASADA technologies/MINER integration is to achieve the ultimate goal of DASADA, which is for the dynamic assembly of MINER in a predictable manner. Integration benefits will include:

- Being able to replace system components through the use of the tool kit and the ADL

- Facilitate the modeling the interaction of components within MINER as well as the external interactions of MINER with other systems

- Allow detection of actual performance and constraint violations using gauges

B. FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS

Since GDIS and SPAWAR have not currently instrumented MINER's system architecture with an ADL to obtain a formal architectural model the Microsoft PowerPoint generated schematics of MINER's architecture were used, which was provided by the MINER's system engineer and programmer. Additionally, input from GDIS technical staff to assist in conducting the assessment of the system requirements was received. Due to the proprietary nature of this material, MINER's functional requirements were gleaned from official GDIS white papers without divulging any of the proprietary issues. The non-functional requirements were actually generated based on the thesis group's level of knowledge of software engineering and usability engineering. The assessment of the DASADA system functional requirements (SR) for MINER is as follows:

- SR1 Main1. Through the use of the ADL capture component interaction behavior of MINER

- SR2 Main2. Through the use of the ADL construct an architectural model of MINER

- SR3 Toolkit1. Through the use of the toolkit replace data sources in relation to architectural model and component interactions

- SR4 Toolkit2. Through the use of the toolkit be able to use technology refresh in relation to architectural model and component interactions

- SR5 ADL1. Through the use the ADL be able to detect component incompatibility in relation to architectural model and component interactions

- SR6 ADL2. Through the use of the ADL execute component replacement in relation to architectural model and component interactions

46

- SR7 T1. Through the use of TBASSCO technology provide design gauges to MINER in relation to architectural model and component interactions

- SR8 T2. Through the use of TBASSCO technology provide run-time gauges in relation to architectural model and component interactions

- SR9 T3. Through the use of TBASSCO technology provide constraint violation analysis in relation to architectural model and component interactions

Additionally, the assessment of the DASADA system nonfunctional requirements is as follows:

- Usability

  - Minimal training time keeping with a steep learning curve

  - Task times are immeasurable at this time but should be less than current levels

  - Adhere to the GUI standards published by Microsoft Windows NT environment

- Reliability

  - Availability: proposed maximum availability of 99.9%

  - Mean time between failures (MTBF): proposed in terms of years. One year being the least acceptable level

  - Mean time to repair (MTTR): may only be down for minutes after it has failed

  - Accuracy: precision and accuracy must be at 99.9%

  - Maximum bugs or defect rate: 1 in 100,000 lines of code is the maximum acceptable level

  - Bugs or defect rate: No critical or significant bugs acceptable

- Performance

  - Enhance current system response time

  - Should not affect throughput or capacity

  - A minor level of degradation of performance is acceptable as long as a increase in reliability and predictability is realized

  - Minimal impact on resource utilization

- Supportability

    - DASADA project will contain a built-in maintenance support capability

UML Use Case diagrams have been used to model the system functional requirements. Table 4.1 lists the five Use Case diagrams with a short description as well as the involved actors.

Name	Description	Actor(s)
System_Overview	Overview of the DASADA technology integration into MINER	Miner_System, toolkit, ADL, TBASSCO
Main	Shows the behavior between MINER and ADL	Miner_System, ADL
TBASSCO	Demonstrates the use of TBASSCO technology	TBASSCO
ADL	Demonstrates the use of ADL technology	ADL
Toolkit	Demonstrates features provided by the toolkit	toolkit

Table 4.1.   Use Case Model Survey.

Table 4.2 lists the four actors, which are used in the Use Case diagrams.

Actor Name	Description
Miner_System	Software application that provides knowledge and information management services
toolkit	DASADA technology capability which will be used to analyze and manipulate the MINER system
ADL	Modeling language used in analysis and for MINER component development
TBASSCO	DASADA technology used for design and performance gauges

Table 4.2.   Actor Survey.

48

Figure 4.1 depicts the Use Case Diagram for MINER/DASADA Technologies.
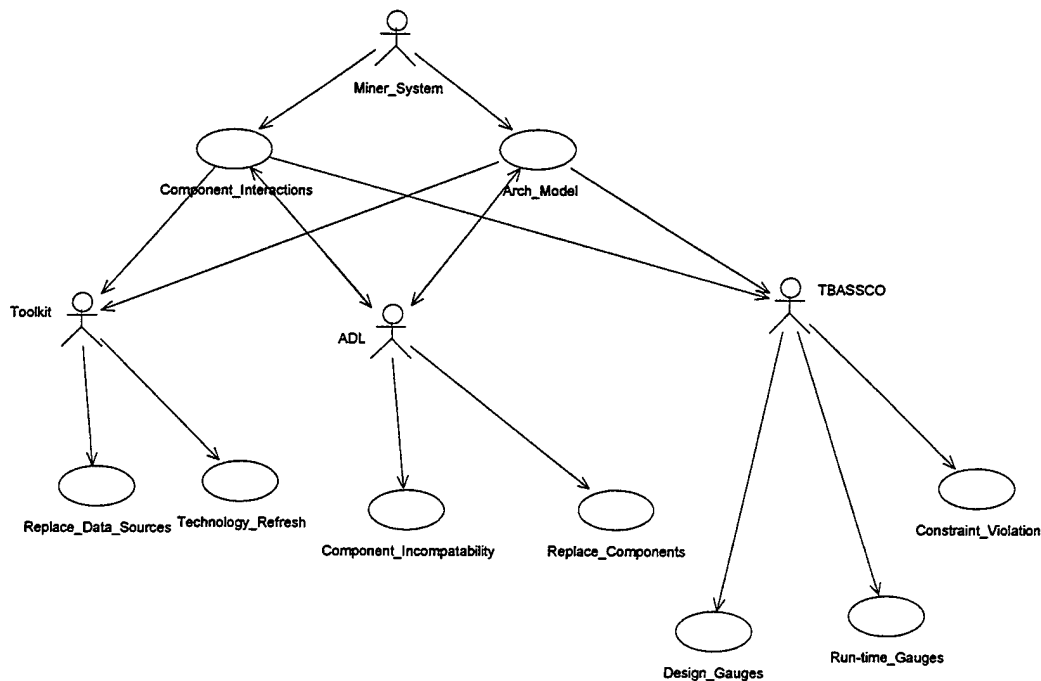
Figure 4.1.   MINER/DASADA Technologies UML Use Case Diagram.

C.   ARCHITECTURAL MODEL ANALYSIS

Since an informal architectural model (Microsoft PowerPoint generated schematic of MINER's architecture, Figure 4.2) was used to conduct the proposed placement of the DASADA technologies' probes and gauges it was determined that it was crucial to the validity of the analysis that the MINER's system engineer be consulted.   During the SPAWAR site visit the question was asked where the MINER's system engineer would likely install DASADA technologies' tools to demonstrate their utility as well as to enhance the performance, reliability, and provide ease of technology refresh. Additional information on the use of DASADA technologies for the enhancement of MINER from GDIS's MINER DASADA EDP White Paper was extracted.

49
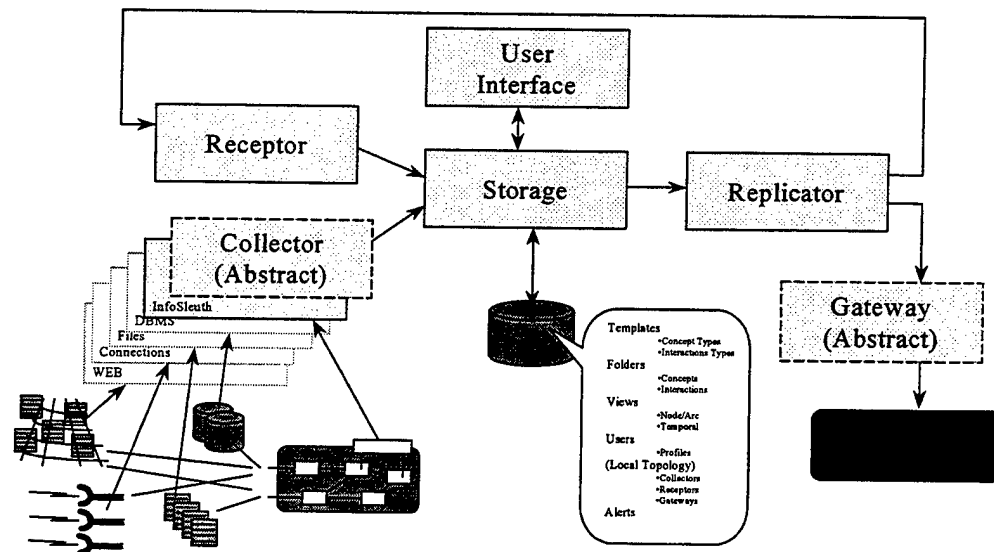
Figure 4.2.   MINER's Architectural Model.

The first planned area of DASADA technology to be used is the ADLs and design-time gauges. The DASADA ADL tools would be used to model the baseline MINER system, the components within the system, and the component connectors. This model will serve as the baseline for the EDP. Additionally, any proposed new components would also be modeled using the ADL to determine how well they would fit into the existing system. Modeling the behavior of the abstract system components and connectors will allow for the replacement of an old component with a similar but different new component while at the same time providing a significant degree of assurance that the new system configuration will continue to function at least at the previous levels of performance and reliability. In conjunction with using the ADLs, the DASADA developed design-time gauges would be utilized to model predicted system performance.

50

The second planned area of DASADA technology to be used is the run-time gauges being developed by DASADA for monitoring actual performance of MINER. Due to the diversity of the components that comprise MINER, the ability to integrate this new gauge technology into the existing system while at the same time ensuring that the required constraints of the different components are adhered to will be closely investigated during the EDP. Additionally, this capability will provide validation of the model, which was created earlier.

The final planned area of DASADA technology is to achieve the ultimate goal of the EDP, which is to investigate the promise of DASADA technologies to enable the dynamic assembly of systems. MINER is ideally suited for this type of experimentation. MINER already uses ontologies to describe the information that it manages which is quite similar to the information provided by the ADL model. Given the information provided by the ADL representation of the model as well as a representation of new components, it is quite possible that the MINER system could use this information to dynamically reconfigure itself once a new component was located from a trusted site and inserted into the existing model. This capability could be used to both replace existing system components, as well as augment MINER functionality with new components and thus would greatly enhance the ability to provide a smoother transition for technology refresh.

GDIS and SPAWAR conducted a preliminary survey of the DASADA technologies, and decided that the work being performed at USC-ISI for TBASSCO as being the most promising for the proposed EDP. TBASSCO's creation of a metadata

again be conducted to assess reliability and usability to determine if DASADA technologies are truly beneficial.
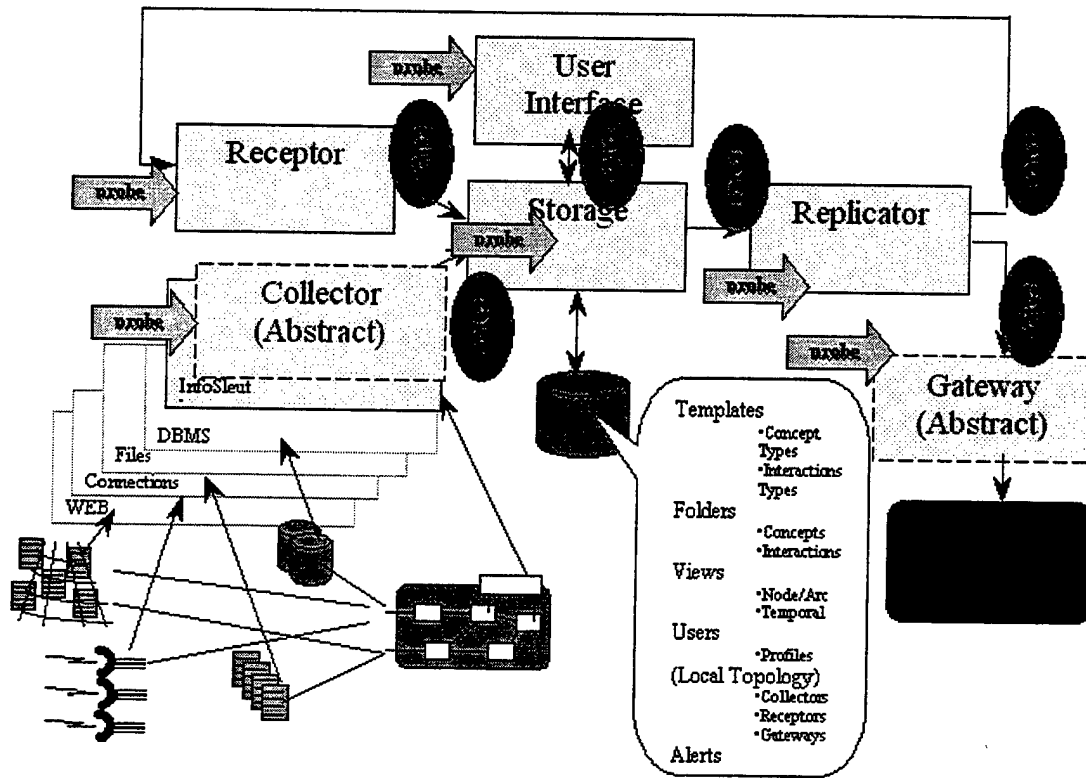
Figure 4.3.   Instrumented Architectural Model.

To achieve the above stated assessments, quantitative measurements need to be established to assist in the evaluation of the effectiveness of applying DASADA technologies to the problem, as well as evaluating the impact of DASADA technologies upon overall system performance. To that end, GDIS has proposed gathering the following measurements for quantitatively evaluating system performance before and after adopting DASADA technologies into MINER:

- Network traffic/latency/throughput
- CPU utilization
- Storage utilization

- Memory utilization

Usually, the very act of monitoring or gauging run-time system performance itself impacts the performance of a system. From the information on the DASADA gauges it is understood that their implementation will generate "events" that will flow through the system and therefore what needs to be evaluated is the impact of these "events" upon overall system performance. The resultant set of measurements will be of great value in ascertaining the impact of inserting DASADA run-time gauges into the existing configuration of MINER.

Additionally, metrics will be kept to evaluate the level of effort required to implement DASADA technologies. Level of effort will be recorded for the following:

- Time to model system
- Time to implement/install gauges
- Time to model/replace components

By tracking the above information, it will be possible to provide benchmark data on the impact to the system development effort of applying DASADA technologies. The resultant data points are only useful if an analogous set of metrics is gathered for system development activities performed without the benefit of DASADA technologies.

F.   TECHNOLOGY TEMPLATE

Based upon the MINER/DASADA technology application findings, a template was designed that can be used to model all of the DASADA technologies. The template components are: target system's functional requirements, target system's nonfunctional requirements, target system's architectural model, analysis of the architectural model, and the specified DASADA technology set of tools. This is done by the use of the following approach: using UML to model the system's functional requirements, software engineering requirements solicitation methods (i.e. storyboarding, brainstorming,

organizational functional requirements toolkit--if available, requirements workshop, etc.) to determine the nonfunctional requirements, use of an ADL to obtain the system architectural model, and the analysis of the architectural model. After conducting these procedures, monitor the application of the chosen DASADA technology set of tools (probes and gauges).

The key to the template is the use of UML to construct a model that will determine the target system's functional requirements. Again, this template can be utilized for any of the DASADA technologies. A checklist was developed to enable anyone to utilize the template. The following are the steps required to determine whether the DASADA technology will meet the objectives they claim to produce, assisting DoD vendors in the selection of the specific technology they require:

- Construct a UML model (preferably with Rational Rose™)
    - Put the DoD technology as the top use case actor, but do not functionally break down the system. It is not necessary
    - Determine the functional requirements you desire and show how they interrelate with each other
    - From your requirements, model your target end states and show any relationships with one another
    - Model an ADL as a use case actor and show its required end states and relationships
    - Model the DASADA technology as another use case actor and show its relationships with its target end states, most likely the probes and gauges that will be applied to the system.
    - Model the relationship between the DoD use case actor and the DASADA use case actor. Even though in our example we only showed the overall UML Use Case diagram, individual relations were modeled in separate diagrams. Additionally, UML class diagrams can be used to provide more detailed information of the involved relations.
- Use software engineering requirements solicitation methods to determine non-functional requirements.

55

- Use of an ADL to determine and analyze your system architecture.

- Determine the placement of probes and gauges based upon the ADL architectural model as well as the identified functional and non-functional requirements.

- Prior to installation of the optimal set of probes and gauges, ensure that the system's baseline levels (performance, reliability, usability, etc.) have been assessed.

- Integrate the set of probes and gauges into your system and then apply your specific system metrics to assess if the DASADA technology enhancements are truly beneficial.

V.   ANALYSIS OF THE SPONSORED PROJECTS PLANS FOR DEMONSTRATION/IMPLEMENTATION

DASADA technology members held their first annual "Demo Days" on June 4-5, 2001 at the Radisson Hotel in Baltimore, MD. It was during this demonstration phase that an early evaluation of the technologies was conducted.

Each organization sent out a description on how their respective "demo" would be conducted, describing the technology used as well as the placement of probes and gauges within each system. The following paragraphs contain the vendor's descriptions as well as a comparative analysis of each technology demonstrated at the exhibition (DASADA, 2001):

A.   GEORGIA STATE UNIVERSITY

Georgia State University intends to demonstrated the following facets of MesoMORPH:

- WorldView ontology capture and display tool -- models concepts, operations, and associations between concepts

- Conceptual gauges -- modeling tool based on semantic networks that determines conceptual distance and other measures for analysis

- ContextView context modeling and display tool -- allows user capabilities (and disabilities) to be described using the HAS-L (XML-based) representation. Also incorporates situational factors (such as low visibility, mobility) and activity factors (what is the user doing with the system in the context).

Tools will be demonstrated using a pilot example (the adaptation of a digital music system to mobile, low-vision, and low-selection accuracy environments) as well as through participation in the Intelligauge Technology Integration Experiment (TIE) group's GeoWorlds target system demonstration.

1. Analysis

During this demonstration, this group did not meet pre-demo objectives as listed in their literature, specifically not integrated with GeoWorlds. Currently, this group has not identified working with any DASADA EDP. An evaluation of this system indicated this group did not demonstrate any applicability of this technology with respect to the DASADA RFP.

B. KESTREL

Kestrel intends to demonstrate basic capabilities of EPOXI in two dimensions:

- First, they will translate architectures from Acme into EPOXI, and then use EPOXI to provide richer semantics to the architecture and to refine it

- Second, they will show the ability to dynamically assemble a consistent system by exploiting specification-carrying code. In particular, Kestrel will show the automated construction of a connector between two net-based components (e.g., buyer and seller agents) based on analysis of their service specifications. The connector embodies an interaction protocol with generated data translators. Any properties that cannot be assured at system design-time are embodied in execution-time gauges.

1. Analysis

This group demonstrated a new approach to the application of ADL's by developing an alternative to the current standard used in most of the DASADA technologies. Their EPOXI technology provides algebraic specification modeling as well as behavior modeling through the use of abstract state machines. This technology has recently come to a point of maturity where they are now looking for a DASADA EDP. An evaluation of this system indicated that this technology has great potential as an architectural modeling language, but it is also recognized that the software engineering community operates within the existing ADL paradigm and therefore Kestrel has a significant challenge to overcome.

C. OBJECT SERVICES AND CONSULTING, INCORPORATED (OBJS)

OBJS intends to demonstrate Version 1.0 of Software Surveyor, which uses:

- Application-specific probes (AppliProbes)
- Generic probes dynamically attached to application components
- Environmental probes (EnviroProbes)

Software Surveyor is a profiling toolkit used to dynamically deduce and render the run-time configuration and behavior of evolving, component-based software. Software Surveyor requires limited a priori knowledge of application connectivity, which makes it possible to use with applications where either full design specifications are unavailable, or the application dynamically reorganizes itself as demands change, new resources become available, and resources fail.

During the demonstration probes will gather information about GeoWorlds, combine the information into a picture of application connectivity and behavior, and highlight anomalies based on comparisons of observed behavior, specified behavior, and prior executions.

1. Analysis

This group demonstrated the execution of both AppliProbes and EnviroProbes at design-time as well as run-time. This technology utilized GeoWorlds to demonstrate its ability to act as diagnostic tool for the system. On the downside, it is limited to Internet-based systems only; it will not work on embedded systems. An evaluation of this system indicated this technology due to its limited scope, is going to have restricted applicability in the DASADA program.

D. SRI INTERNATIONAL

SRI International intends to demonstrate a gauge that measures a fault-tolerance property. Specifically, the fault tolerance property is the number of failures of components that contribute to mission-critical functions that can be tolerated without loss of critical system functionality of an evolving system.

timing analyzer to gauge schedulability, UNCLE to gauge constraint consistency, and QRAM to gauge optimal resource allocation.

Dynamic adaptation of on-board situational awareness is the process of reconfiguring the on-board computing resources to maximize use of the available intelligence/sensor sources. It is envisioned that mobile code will be shared between the sensor and shooter platforms to facilitate the use of the sensor data. Slack scheduling is used to gauge the schedulability of this mobile code.

1. Analysis

Northrop Grumman is fortunate to be able to claim that their technology deploys today on operational B-2 Spirit aircraft. A mission-planning tool used aboard the B-2 is being utilized to deploy gauges that measure run time and compile time compliance to the architectural model. An evaluation of this system indicated Northrop Grumman is ready to move on to the next phase of the DASADA program.

F. UNIVERSITY OF SOUTHERN CALIFORNIA INFORMATION SCIENCES INSTITUTE (USC/ISI)

USC/ISI intends to show in their demonstration how the SIM-TBASSCO metadata framework supports semantic-level gauges that help application developers identify and combine interoperable software components. This facilitates rapid composition of semantically assured software architectures as components are assembled into special-purpose programs. USC/ISI will show how their semantically based scripting tool helps users design a data-flow style architecture at multiple abstraction levels, and also how it helps users to incrementally modify, instantiate, and test the architecture by allocating correct resources. During the demonstration, USC will show how scripting gauges can help users easily identify semantically interoperable and compatible software

components. In addition, they will demonstrate how the component insertion gauge can help system engineers measure the semantic interoperability and compatibility levels of a new software component prior to integrating it into the system.

The goal of the SIM-TBASSCO (Semantic Interoperability Measures: Template-Based Assurance of Semantic Interoperability in Software Composition) project is to develop a metadata framework for describing software components to support the dynamic assembly of software systems. As a test bed application for this work, SIM-TBASSCO has adopted GeoWorlds, a component-based Web and geographic information management system.

1. Analysis

This technology is at a mature level and is being utilized by several other DASADA groups as its respective EDP. This group demonstrated several different views such as application developer, system administrator, and component developer for design time gauges. This group is coordinating with Columbia University for the inclusion of run-time gauges. The developers of GeoWorlds are assisting in the actual development of SIM-TBASSCO. An evaluation of this system indicated this group is one of the most mature of the DASADA technologies and is ready to move on to the next phase.

G. BBN TECHNOLOGIES

BBN Technologies intends to demonstrate an adaptive "meta-search engine" to illustrate the use of the following technology:

- Robust Workflows for Distributed Workgroups

BBN provides a workflow technology to robustly organize distributed services across broad range of operating contexts, environments, and connectivity profiles. The BBN "Service and Contract" (S+C) solution is a task-based workflow implementation to

specify, compose, invoke, monitor, and adapt the organization of distributed services (components and gauges) within a dynamic operating environment. The S+C workflow provides a mechanism to solve "cross-cut" service constraints across distributed nodes. This permits design of workflows that can balance diverse and interdependent measures of performance ranging from component, application, and system properties. The "Service and Contract" workflow framework is externally accessible via XML (for integration and visualization, etc).

XML import/export capability enables transformations of workflow models into range of export/import representations (Architecture Description Languages, etc): integration and monitoring.

1. Analysis

This group's technology demonstration included a web-based diagnostic tool for quantifying Internet search engine results and is currently coordinating with OBJS and Columbia for future development partnerships in addition to talking to SPAWAR about the "Habitats" project. An evaluation of this system indicated the technology has the potential to become part of the unified toolset for web-based systems. BBS currently does not have a DASADA EDP, but it could work well with SPAWAR's MINER project.

H. CARNEGIE MELLON UNIVERSITY (CMU) / IMPACT

The CMU IMPACT (Integrated Methods for Predictive Analytic Composition and Tradeoff) demonstration will feature a visual demonstration of dynamic assembly and analysis technologies applied to a surveillance and tracking challenge problem in avionics platforms. The demonstration presents a methodology for designing and scheduling a radar RF timeline to maximize tracking quality for a variety of dynamic mission scenarios. The overall objective of the demonstration intends to showcase a

preliminary set of technologies that supports dynamic assembly and rapid assessment of high assurance, resource constrained systems. The demonstration uses a version of the F-16 Falcon-Star avionics simulation environment that provides realistic, in-context stimulus for demonstrated algorithms and techniques. Displays will show the quality of the tracking achieved in the presence of mission driven dynamic system loads. In addition, the demonstration will provide visualization of computing resource allocation decisions as reported through run-time gauges embedded in the application.

Three tools associated with the major application demonstration will be presented:

- TimeWiz -- a comprehensive real-time system design tool
- Visual Q-RAM -- QoS-based resource allocation model
- Visual RTQT -- tool to visualize real-time queuing behavior or applications

1. **Analysis**

This group demonstrated all proposed objectives from the DASADA literature. This group works closely with Lockheed Martin on real time scheduling and context testing on the F-16 avionics platforms. CMU is doing breadboard testing and creating prototypes for a new advanced avionics suite proposed for future aircraft development. An evaluation of this system indicated this group is ready to move on to the next phase of the DASADA program.

I.   TEKNOWLEDGE

Teknowledge intends to demonstrate the use of an Acme architectural style (via PowerPoint-based Design Editor) to:

- Design dynamic system configurations
- Deploy probes to instrument that dynamic configuration
- Display an animated visualization of that architecture as well as selected non-functional properties of its components

- Manually reconfigure that deployed system during its execution

The configuration animation will also highlight departures from an UML-based simulation of the system's nominal behavior.

1. Analysis

The Teknowledge group demonstrated technologies not stated in the pre-demo literature. Two gauges were presented; one was an architectural probe that provided information for process analysis; the other was a security probe and gauge set called "SafeMail", which analyzed e-mail run time virus behavior vice a traditional anti-virus software program, which analyzes the virus signatures. This probe and gauge set is advertised to run on any Windows NT 4.0 and Windows 2000 based e-mail program. This technology is currently being employed at DARPA Advanced Technology Office (ATO) and will soon ship to USPACOM as well as the Software Engineering Research group at the Naval Postgraduate School. An evaluation of this system indicated this technology is applicable only to web and network-based systems, and is not applicable to embedded software systems.

J. UNIVERSITY OF MASSACHUSETTS

The University of Massachusetts intends to demonstrate how its technologies, working in concert with technologies from Honeywell Technology Center, and CMU to support software adaptation. The UMass demonstration hypothesizes that a helicopter is performing a training mission when contingencies arise forcing the need to change mission in mid-flight, first because of weather conditions and then because of an urgent search and rescue request. The demonstration scenario is as follows:

- The first mode change involves a predetermined software configuration that was proposed and validated during design-time. The design process is defined and executed as a Little-JIL process, which in this case employs MetaH to specify the architectural specification, which in turn is analyzed

by UNCLE for consistency constraints, by QRAM for timing constraints, and by FLAVERS for behavioral requirements. MetaH generates the corresponding software system from the resulting validated architectural specification, thereby supporting one of the possible predetermined helicopter mode changes.

- The second mode change is not predetermined and, in this case, an on-board Little-JIL process oversees a dynamic software reconfiguration that employs MetaH slack stealing, limited resource re-assignment, and rapid, time-permitting re-analysis. Based on gauge readings and pilot direction, a new "safe" configuration is selected and reported back to the base station where off-line analysis continues to evaluate if a more effective alternative exists, while the helicopter continues on its newly defined mission.

Four different University of Massachusetts technologies are used in this demonstration:

- Little-JIL/Juliette -- a process definition language and execution system
- Midas -- a resource specification and management system
- FLAVERS -- a finite state verification system
- Propel -- a system for eliciting correct software properties

1. **Analysis**

This group decided not to use parts of its demonstration plan due to integration difficulties. This was a conscience decision on the part of UMASS and Honeywell. What were demonstrated were actually four separate demonstrations, one for each UMASS technology. Each system performed as advertised, showing the maturity of the technology to be applied to an EDP. UMASS is currently working with AMCOM to try to find an EDP. The Theater High Altitude Air Defense (THAAD) program has shown keen interest in FLAVORS specifically. An evaluation of this system indicated that once an EDP is identified, these technologies would quickly prove themselves worthy of further analysis into the next DASADA phase.

K. UNIVERSITY OF OREGON

University of Oregon researchers intend to demonstrate technologies for extracting run-time gauges from design-time models. A key feature of their approach is

providing gauges with a "yellow zone," which indicates potential trouble while corrective or ameliorative action remains possible. The approach will be illustrated through an experimental application to a problem provided by NASA.

1. Analysis

This group did not execute its intended demonstration events. Instead, University of Oregon demonstrated a new technology called GenSet, which is a scriptable tool for information fusion. This tool gives you design information for reverse engineering. The literature describes GenSet as "an early, fragile prototype. It is incomplete and not ready for outside use." Because of this, University of Oregon is not close to working with a DASADA EDP.

The second demonstration on Finding the Yellow Zone was a jury-rigged scheduling elevator simulator, which lacked a scheduling algorithm to properly execute the program. This had no relevance to the DASADA program and was a complete waste of time and energy.

An evaluation of this system indicated that GenSet might have some "future" potential as a reverse engineering tool. Unfortunately, University of Oregon's literature indicates the technology is immature and not ready for any near-future implementation into a DASADA EDP.

L. COLUMBIA UNIVERSITY/WPI

Columbia University and WPI intend to demonstrate how Kinesthetics eXtreme (KX) using the specific example of GeoWorlds as the target system:

- Probes a target system using Active Interfaces for automated source code instrumentation

- Analyzes streams of partially ordered events for distributed and time-based patterns, potentially indicating faults or undesirable conditions, using our XML-based Universal Event System

- Displays continuously updated visual gauges and potentially other analysis tools through our TRIKX portal framework

- Reconfigures the running system using our Gaugent variant of Worklet mobile agents for process-aware systems

Columbia will also attempt to demonstrate on-the-fly reconfiguration of the KX system itself based on Flexible XML (FleXML) schema composition capabilities and Workgroup Cache intelligent information propagation system.

WPI's Active Interface technology provides a mechanism to collect information about running software systems. Their demonstration will consist of two parts:

- Use of the Active Interface Development Environment (AIDE) compiler to instrument Java source code with hooks that deliver accurate, timely information to a gauge notification infrastructure

- Replacement of a GeoWorlds Library with an Active Interface enabled version and demonstrates probe deployment and execution using the Active Interfaces Probe Run-Time Infrastructure

The second portion of the demonstration will showcase a number of gauges, as well as the associated probes that provide the gauges pertinent raw data:

- Experience-Based Expectation gauges that monitor the time required for a remote service (or a series of remote services) to perform a task. These gauges will keep track of past performance and will flag services that do not meet their expectation.

- Failure isolation gauges that work by pairing before and after events for important method invocations in a target system will also be demonstrated. These gauges are also useful to determine possible sources of failure.

- Domain-specific gauge designed to emit events when pre-specified conditions are violated will be showcased.

All of the above mentioned probes and gauges would be used to monitor the GeoWorlds target system.

1. Analysis

This group demonstrated the ability to analyze source code by targeting specific connector or components and then showed how they can monitor the actual run-time performance of the code while the application is executing. Additionally, Columbia/WPI demonstrated the ability to animate the source code into a virtual reality model, allowing navigation through the model to view specific connectors and components. All the software is written in Java 2 programming language, which has potential for use in web and network-based systems, but not in embedded systems. Utilizing GeoWorlds, this technology has a platform in which it can operate its probes and gauges. An evaluation of this system indicated this group seems to have some merit for further consideration.

M. THE UNIVERSITY OF COLORADO

The University of Colorado intends to demonstrate the following:

- Scalable Publish/Subscribe Communication (Siena) -- The University of Colorado has developed Siena, a publish/subscribe service whose goal is to support large-scale communication in a wide-area network which provides flexibility in connecting heterogeneous, distributed systems. Siena is being used by DASADA researchers as a common event notification mechanism for probes and gauges.

- Automated Configuration and Deployment (Software Dock) -- The University of Colorado has developed Software Dock, which is an agent-based system, to support advanced configuration and deployment scenarios. An additional tool has been developed for analyzing the possible configurations of software systems for early detection of configuration errors.

- Information Integration Environment (INFINiTE) -- The University of Colorado is examining dynamic and adaptable techniques for automating support to meet the challenge of discovering and managing the relationships among software artifacts. INFINiTE is a Web-based environment for automatically generating relationships between software artifacts via the use of software agents, known as integrators, storing them using open hypermedia, and making them available within the software artifact's original editing environment.

1. Analysis

The University of Colorado is utilizing GeoWorlds as a test bed for Siena and INFINiTE, as well as the implementation of a fitness gauge. Although the discussion includes the use of GeoWorlds as their test bed, this was not demonstrated. The technologies appear to be at a maturity level where they could be instrumented into GeoWorlds at any time. An evaluation of this system indicated this technology appears to have applicability in a web-based and network-based environment only. It does not appear it would work well with embedded systems.

N. CARNEGIE MELLON UNIVERSITY (CMU) / RAINBOW

The CMU/Rainbow project (a.k.a., Architecture-based Adaptation of Complex Systems) intends to demonstrate the following:

- Ability to monitor performance characteristics of an executing system
- Ability to interpret these characteristics in the context of software architecture

In the demonstration, Remos (a network bandwidth service used to probe the bandwidth being received by an application) produces network bandwidth information that is interpreted as architectural properties by AcmeStudio (a software architecture design tool used to design and visualize a software architecture). Additional demonstrations will involve translation from Acme to UML and xArch.

1. Analysis

CMU/Rainbow has not identified any DASADA EDP, although other DASADA technologies are utilizing their development tools (i.e. AcmeStudio). The demonstration modeled how the system architecture was performing (i.e. with respect to bandwidth, compression, and file size). On the downside, the system has a significant amount of initial setup overhead. An evaluation of this system indicated this technology looks

promising, but the level of utility cannot be determined without taking into consideration how their tools are used in other DASADA projects.

O. UNIVERSITY OF SOUTHERN CALIFORNIA CENTER FOR SOFTWARE ENGINEERING (USC/CSE)

USC/CSE intends to demonstrate three related capabilities:

- First demonstrated capability is a lightweight, extensible architecture-based software implementation infrastructure. The infrastructure allows application modeling in terms of software components, connectors, and messages. It also inherently supports placement of gauges at arbitrary locations in the architecture to monitor its run-time behavior.

- Second demonstrated capability will leverage the infrastructure in the implementation of special-purpose software connectors for ensuring application reliability during component upgrades

- Third demonstrated capability will augment the implementation- and run-time support of the first two capabilities with design-time modeling, analysis, and system generation support that combines the power of static modeling (i.e., pre- and post-conditions) and dynamic modeling (i.e., state charts) techniques

1. Analysis

This group did not demonstrate its intended capabilities with any EDP, although they did provide a highly scripted war game demonstration. Using several different PDA's and one laptop, they attempted to network these devices, which when executed, experienced several errors. A pure theoretical demonstration of their DRADEL toolset was given, although applicability into the DASADA program is suspect. An evaluation of this system indicated this technology has no merit in the DASADA program.

P. GEORGIA INSTITUTE OF TECHNOLOGY

Georgia Institute of Technology, with subcontractor Michigan State University, leads the DYNamic Assembly from MOdels (DYNAMO) project, which is concerned with automating the process of producing high-assurance assemblies built from independently constructed software components. This DYNAMO demonstration will attempt to illustrate three distinct points of view:

71

- From the view of the manager of a product line, they show how components can be selected to comprise an assembly

- From the view of a component designer, they show how component properties can be specified graphically and static analyses performed on them

- From the view of the end user, they show the resulting assembly executing, together with gauges depicting dynamic system properties

1. Analysis

This demonstration had no scenario and was admittedly "canned". This group was able to show their three viewpoints, but only with a static representation. The group "hopes" to work with SPAWAR in the future. An evaluation of this system indicated this technology is in an immature state and should not be considered for further evaluation in the DASADA program.

Q. HONEYWELL TECHNOLOGY CENTER

Honeywell will be providing two demonstrations that illustrate key technologies:

- The first being the UNCLE system which will demonstrate the use of set-wise constraints as design-time gauges for verifying high-level properties of a system of systems (e.g., helicopter system architectures). To demonstrate the feasibility of integrating the UNCLE infrastructure with an external solver, the solving of these constraints will be done using a constraint engine built in SICStus Prolog;

- The second being how work on slack servers can provide, increased throughput of real-time gauge readings to distributed interactive non-critical applications while simultaneously supporting safety-critical applications. To show this, Honeywell plans to compare the throughput rates of three IP communication channels between NT processes and embedded MetaH processes executing while co-hosting a (simulated) resource-constrained safety-critical process. The MetaH executive will feature three communication server-scheduling disciplines: background, periodic polling, and a slack server.

1. Analysis

This group gave a presentation on the merits of slack scheduling, which their product, MetaH addresses. MetaH has been in development for over 10 years, with refinements and upgrades ongoing. MetaH was used in 1997 on another small DARPA

project, which introduced slack scheduling into aircraft avionics packages. At this time, there is no platform integration planned for MetaH, although this is a proven technology, which has the potential to work on real-time systems such as advanced avionics suites or integrated combat systems. An evaluation of this system indicated this is a viable technology that needs to find a DASADA EDP.

R.   UNIVERSITY OF CALIFORNIA, IRVINE

All demonstrations by UC Irvine will take place in the context of the Airborne Warning and Control System (AWACS) command and control radar surveillance system. Subcontractor Lockheed Martin Aerospace Systems provides data processing subsystem solutions for diverse platforms including the E-3 AWACS aircraft. UCI intends to demonstrate how their technologies can be applied to help in the rapid exploration of alternative architectures for AWACS. Specific technologies to be demonstrated are xADL 2.0, ArchStudio 3.0, and ArchDiff:

- xADL 2.0 -- an XML-based architectural representation
- ArchStudio 3.0 -- an architecture tool suite
- ArchDiff -- an architecture-differencing tool

1.   Analysis

This group was able to describe the various run and design-time gauges they are planning to employ on the E-3 AWACS aircraft Block 40-45 software upgrade to its tracking and identification system. UCI has developed its own design-time gauges for the upgrade, and is relying on Lockheed Martin to produce the run-time performance gauges required for this system. This is the only group to demonstrate a Human Computer Interaction (HCI) approach to the design and implementation of gauges. An evaluation of this system indicated that UCI is ready for the follow-on phase of

DASADA. UCI presented the most professional and thorough demonstration of all the DASADA project groups.

S.   VERIDIAN PACIFIC-SIERRA RESEARCH (PSR)

Veridian Systems intends to demonstrate terrain-reasoning software being reconfigured via the Venice tool. The demonstration will show the ability to make a request of the terrain server to generate a terrain product. The server will then be reconfigured, via a web browser, to generate the same product using a higher fidelity algorithm requiring higher fidelity data. This demonstration will also be incorporated into the Intelligauge TIE GeoWorlds system demonstration, providing a terrain product that can be displayed on the GeoWorlds map.

1.   Analysis

This group was able to demonstrate at design time to compose a component and then insert that component into a software subsystem. PSR could then execute that subsystem in a run-time environment to obtain its results. Utilizing GeoWorlds as its EDP, this technology can only be used in a web-based environment and cannot be used with embedded systems. An evaluation of this system indicated this system has tremendous ability to dynamically configure system components, although it is limited to web-based systems.

VI. CONCLUSIONS AND RECOMMENDATIONS

During the writing of this thesis it quickly became apparent that an in-depth analysis of the 19 funded DASADA projects could not be completed and therefore the scope of the problem space was reduced in order to deliver a quality product. The best way to approach the problem was to conduct a limited but comprehensive research on all of the projects so that a thorough assessment of their potential contribution to the overall DASADA Program goal could be determined. Additionally, an assessment of a sufficient development rate with an EDP to demonstrate their capabilities was required.

During the research phase several individual projects were queried on their current state of development. Information on the progress of the DASADA projects was not forthcoming making an in-depth assessment difficult. This restriction was detrimental to the research effort, limiting any further research to the information provided by the program office.

During the fact-finding efforts at the "DASADA Demo Days" in Baltimore, Maryland, a significant amount of insight into the development status of each of the projects as well as comprehensive information into each of the technologies was attained. It was observed that some of the projects were aggressively coordinating with other technologies as well as working with an EDP. Several projects just recently matured their technology to the point where they were going to contact one of the EDPs in the near future for demonstration purposes. Lastly, there was a hand full of projects that were not even close to the development level to demonstrate their projects much less than working with an EDP in the near future.

75

An interview with DARPA ITO's Acting Director, Dr Mark Swinson was conducted on June 6. Keen insight into the DASADA Program was obtained from this interview. DASADA is considered a fringe program because the program's focus is not on real-time or embedded systems as are the other DARPA ITO programs. In fact, Dr. Swinson stated, "How DASADA actually fits into the DARPA ITO arena is up for question". Another issue raised was that there appeared to be a lot of familiar faces from the software engineering community that were now stating that they had mature DASADA technologies, but that the technologies that they are offering are actually existing programs that these research groups had developed in the past. A third issue was that the out-years funding for the program was up for review pending development results; this fact was emphasized by Dr. Swinson saying, "there needs to be some measurable results now, not just three years out because any technology can look good in several years" (Swinson, 2001).

A valuable service was provided to DARPA by the assessment of the 19 projects, but due to the current program management office it is suspected that the information provided would not be used.

Out of the 19 projects, there is only a handful that should be considered for future funding based upon their level of effort over the past several months, as well as their level of technology maturity to be able in the next year to actually provide a component to insert into the DASADA Dynamic Assembly Toolkit. Those projects are:

- Northrop Grumman's Dynamically Adaptable Component-based Data Link Systems (DACDLS)
- USC/ISI 's Semantic Interoperability Measures: Template-based Assurance of Semantic Interoperability in Software Composition (SIM-TBASSCO)

- CMU's Integrated Methods for Predictive Analytic Composition and Tradeoff (IMPACT)

- Columbia University and WPI 's Coping with Complexity: A Standards-based Kinesthetic Approach to Monitoring Non-standard Component-based Systems/Kinesthetics eXtreme (KX)

- UCI's Proteus: Assessment and Adaptation Through Dynamic Architecture Technology

- Veridian Pacific-Sierra Research's Innovative Gauges for Component-based System Assembly

The below listed technologies show great promise but will have to integrate with an EDP in order to provide validation for further consideration in the DASADA Program:

- Kestrel's Specification-Carrying Software

- University of Massachusetts' Process Guidance and Validation for Dependable On-The-Fly System Adaptation

- Honeywell Technology Center's Gauges for Reliable Adaptation (includes MetaH)

Observations obtained during the research of this thesis have determined that the technologies, which are currently coordinating with industry on the development of embedded software systems, are the most applicable to the original spirit of the DASADA Program. This analysis also concludes that there are particular web and network-based systems that in all likelihood will prove to be of considerable benefit to DoD.

There are two aspects of the DASADA program that warrants mentioning due to their success; the first being that DARPA deemed the best method to achieve the program objective was to merge academia with DoD projects so that the developing technologies had readily available real-world projects to demonstrate their advanced technological capabilities. The second aspect being the exposure to DoD engineers of the current state of software engineering practices that the DASADA program exemplifies when the

DASADA Winter Principal Investigator (PI) Meeting was held at Naval Postgraduate School in January 2001.

LIST OF REFERENCES

DARPA, "Dynamic Assembly for Systems Adaptability, Dependability, and Assurance", http://www.darpa.mil/baa/baa00-20.htm, December 1999.

DASADA, "DASADA Events", http://schafer-ballston.com/dasada, January 2001.

Milligan, J., "DASADA Program Project Information Sheets", http://www.if.afrl.af.mil/tech/programs/dasada, January 2001.

Milligan, J., "DASADA Program Overview", http://www.if.afrl.af.mil/tech/programs/dasada, April 2001.

Salasin, J., "DASADA: Overview and Gauge Descriptions", http://schafer-ballston.com/dasada, April 27, 2000.

Salasin, J., "DASADA: Semi-Annual Review", http://schafer-ballston.com/dasada, November 2000.

SchaferCorp, "Proposer Information Pamphlet for BAA 00-20", http://schafer-ballston.com/dasada/final_pip/dasada_pip.html, January 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

Defense Advanced Research Projects Agency, *DASADA Projects*, http://schafer-ballston.com/dasada, January 2001.

Fowler, M. and Scott, K., *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 2d ed., Addison-Wesley, 2000.

General Dynamics Information Systems, *DARPA DASADA MINER EDP White Paper*, Proprietary, 27 October 2000.

General Dynamics Information Systems, *MINER as Desktop Knowledge Manager White Paper*, Proprietary, 9 March 2001.

Hofmeister, C., Nord, R., and Soni, D., *Applied Software Architecture*, Addison-Wesley, 2000.

Leffingwell, D. and Widrig, D., *Managing Software Requirements: A Unified Approach*, Addison-Wesley, 2000.

Peters, J. and Pedrycz, W., *Software Engineering: An Engineering Approach*, John Wiley & Sons, Inc., 2000.

Shaw, M. and Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.

THIS PAGE INTENTIONALLY LEFT BLANK

DASADA GLOSSARY

ABASs – Attribute-Based Architectural Styles

AC – Analytic Composability. To compose analyzable models from sub-models using formal rules

Acme – Architectural representation/interchange tools

ADL - Architectural Description Language

AFRL – Air Force Research Laboratory

ALP – BBN Technologies' Advanced Logistics Planning architecture. It is a scalable, distributed architecture that fully automates the logistics process in support of a large-scale, globally deployed enterprise

AMCOM – Army Aviation and Missile Command

ATD – Advanced Technology Demonstration

Aura – Task management system

CSCI – Computer Software Configuration Item

CORBA - Common Object Request Broker

COTS – Commercial Off-The-Shelf software applications, hardware components, C2-Style Architectures – UCI's component-based and message-based architectural style for constructing flexible and extensible software systems. C2 architecture is a hierarchical network of concurrent components linked together by connectors in accordance with a set of established style rules

C4ISR – Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance

C4I – Command, Control, Communications, Computers, and Intelligence

DACDLS – Dynamically Adaptable Component-based Data Link Systems

DARPA ITO – Defense Advanced Research Projects Agency Information Technology Office

DII-COE – Defense Information Infrastructure Common Operating Environment

DLL – Microsoft's Dynamic Link Libraries

DoD - Department of Defense

DYNAMO - DYNamic Assembly from MOdels - Joint Georgia Tech and Michigan State project

EDCS – Evolutionary Design of Complex Software program

EDP – Experimental Demonstration Project

FLAVERS - A static data flow analysis system developed by UMASS

Gauge – Software that converts data collected by a probe to a measure that's meaningful for system tuning

GBS – Global Broadcast Service. A broadband broadcast satellite communications system

GDIS – General Dynamics Information Systems

GeoWorlds – Test bed application. Large component-based system in use at PACOM.

Geographic information systems plus web processing

GUI - Graphical User Interface

HTTP – HyperText Transfer Protocol

IEM – Information Enterprise Management

IMPACT – Integrated Methods for Predictive Analytic Composition and Tradeoff. Joint CMU and Lockheed Martin project

InfoSleuth – An intelligent agent-based data acquisition utility program that provides seamless access to heterogeneous information sources used by MINER

IP – Internet Protocol

ITSA – Intrusion-Tolerant Software Architectures

JINI/Java – Distributed system architecture based upon Java programming language, which consists of a programming model and a run-time infrastructure. The programming model helps designers to build reliable distributed systems as a federation of services and client applications. The run-time infrastructure resides on the network and provides

84

mechanisms for adding, subtracting, locating, and accessing services, as the system requires.

Kinesthetics – Refers to natural ability to detect bodily movement and tensions by sensors located in muscles, tendons, and joints. KX technology for continual validation is said to be kinesthetic because it embeds probes within the system elements that compose the software architecture of the system being monitored.

Little-JIL - A process definition and execution language developed by UMASS

MCP – Master Caution Panel

Menage – A representation of configurable architectures, extending traditional architecture description languages to address versioning, variability, and optionality in systems

MetaH – ADL for time critical and dynamic systems

MINER – Managed Information and Network Exchange Router

MTBF -- Mean time between failures

MTTR -- Mean time to repair

NCA – National Command Authority

NFCS – Naval Fires Control System

OS – Operating System

USPACOM – United States Pacific Command

POSETs – Partially ordered sets

Probe – Software that interacts with the operating system to collect data

QoS – Quality of Service factors/constraints

Q-RAM – QoS-Based Resource Allocation Model. A methodology for optimizing application quality of service and supporting design tradeoffs.

QuO – BBN Technologies' Quality Objects research team, which is involved in the integration of the capabilities of distributed object computing (DOC) technology such as COBRA or Java RMI with emerging capabilities that support various sorts of QoS in distributed systems.

Rapide – An architecture description language and tools developed for DARPA by Stanford. Modeling essential complexity in four phases: specification of Rapide architecture; execute with Raptor Engine; analyze generated POSETs; assess invariant satisfaction and constraint violations

Remos – Carnegie Mellon University's run-time monitoring infrastructure

ROSA-D – Rotorcraft Open Systems Avionics Demonstration

SAAGE – Integrated environment for transforming C2-style architectures into UML

Siena – Scalable event notification service used to capture, fuse, and disseminate information in a wide-area network. An example of a DASADA infrastructure tool.

SIPRNET – Secure Internet Protocol Network

Software Dock – An agent-based, distributed infrastructure for describing, deploying, and activating components

SPAWAR – Space and Naval Warfare Systems Command

SR – System functional requirement

TACOM - Tank-Automotive and Armaments Command

TBASSCO – Template-Based Assurance of Semantic Interoperability in Software Composition. Also referred to as SIM-TBASSCO, which stands for Semantic Interoperability Measures: Template-Based Assurance of Semantic Interoperability in Software Composition.

TIE – Technology Integration Experiments

TimeWiz® - A product of TimeSys Corporation. A visual software environment for designing, modeling, and analyzing timing behavior and reconfigurability of systems.

UML – Unified Modeling Language

UNCLE - UNiversal Constraint Language and Engine

VAST-C – Vehicular Advanced Software Technology Consortium

Venice – Web-enabled component infrastructure used for design-time composition

VRTQT – Visual Real-Time Queuing Theory. A tool to visualize and predict the behavior of different scheduling policies in real-time systems.

xADL2.0 – Set of xArch XML schema extensions and libraries (API's) with the primary focus on modeling the design-time composition of a software system

xArch – Extensible, XML-based core of architectural elements with the primary focus on modeling the run-time composition of a software system

XML – Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Suite 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Rd.
   Quantico, VA 22134-5027

4. Director, Marine Corps Research Center 2
   MCCDC, Code C40RC
   2040 Broadway Street
   Quantico, VA 22134-5107

5. Director, Studies and Analysis Division 1
   MCCDC, Code C45
   300 Russell Road
   Quantico, VA 22134-5130

6. Professor Luqi 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943

7. Professor Man-Tak Shing.. 1
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA 93943

8. Professor John Osmundson 1
   Information Systems Department
   Naval Postgraduate School
   Monterey, CA 93943

9.   Professor Richard Riehle 1
      Computer Science Department
      Naval Postgraduate School
      Monterey, CA 93943

10.   Professor Dan Boger...1
      Information Systems Department
      Naval Postgraduate School
      Monterey, CA 93943

11.   Major Wayne S. Mandak 1
      205 Colmar Rd.
      Seaside, CA 93955

12.   LCDR Charles A. Stowell, II..1
      18 Mervine St.
      Monterey, CA 93940